한국인공지능학회
2021 하계 및 추계 학술대회 논문집

하계학술대회: 2021년 7월 8일~10일 (온라인)

추계학술대회: 2021년 11월 4일~6일 (온라인)

한국인공지능학회
Korean Artificial Intelligence Association

한국인공지능학회
2021 하계 및 추계 학술대회 논문집

하계학술대회: 2021년 7월 8일~10일 (온라인)

추계학술대회: 2021년 11월 4일~6일 (온라인)

한국인공지능학회
Korean Artificial Intelligence Association

2021 하계 및 추계학술대회 논문집을 발간하며

한국인공지능학회 회원 여러분,
어려운 여건 속에서도 인공지능 기술 발전을 위해 노력해주신 회원님들의 수고에 깊은 감사를 드립니다.

저희 학회는 2021년 학술대회 논문집을 발간합니다. 학술대회를 두 번 개최하여 논문을 모집했습니다. 7월 8일(목)부터 10일까지는 하계학술대회 그리고 11월 4일부터 6일까지 추계학술대회를 온라인으로 개최하였습니다. 추계학술대회는 2020년 12월 설립된 LG AI 연구원과 공동주최로 개최하였습니다. 투고된 논문은 엄정한 최고전문가 심사를 거쳐서 최종승인되었습니다. 승인된 논문들 중 다수의 논문들은 인공지능 관련 top tier 학회의 논문으로 동시에 승인되기도 하였습니다. 발간된 논문집은 인공지능 기술 정보 공유차원에서 발간되었고 좋은 참고자료로 사용되기를 바라면서 앞으로 출간될 논문에 인용될 수 있으면 좋겠습니다.

회원 여러분들의 많은 성원을 기대합니다.

(사)한국인공지능학회장 유창동

2021 하계 및 추계학술대회 논문집을 발간하며

2021년 (사)한국인공지능학회은 두 차례의 학술대회를 성과적으로 치루어내고 회원을 확대해 왔으며, 인공지능 연구의 최고 전문가, 연구원, 기업인들의 학회로 그 위상을 갈수록 확대하고 있습니다. 코로나 펜데믹이라는 여러운 조건에서도 성과적인 학술대회와 논문투고를 해 주신 많은 연구자분들에게 다시금 감사인사드립니다. 이번 학술대회들에서는 국내 인공지능 연구의 주요 성과들을 교류하고 검토할 수 있는 논문투고 및 승인 과정이 있었습니다. 여기에 참여해서 하계 및 추계학술대회 프로그램 위원으로 활동해 주신 모든 연구자분들에게 다시금 감사드립니다.

구체적으로 말씀드리면 하계학술대회/추계학술대회에 총 76편의 논문이 접수되었고 이 중 68편의 논문이 승인되었습니다. 그리고 승인된 논문 중 8편을 선별하여 본 논문집에 실었습니다. 2021년 학술대회에 투고된 논문들을 보면서 국내의 여러 대학과 기업들에서 참여해 주셨습니다. 인공지능 연구를 선도하는 주요 대학들뿐만 아니라 육사에서도 논문을 제출해주셨고, 기업에서는 LG에서 논문들을 투고해 주셨습니다. 연구 내용들 역시 어느 때보다 수준이 높았으며, 어느 국제학회에서도 경쟁력있는 훌륭한 논문들을 많이 제출해 졌습니다. 다시금 감사드립니다. 선별 과정 상 이런 훌륭한 모든 논문들을 다 출판하지 못해 매우 아쉽습니다만, 저희 학회가 국내 인공지능 연구자들의 교류 확산과 국제경쟁력 제고에 큰 역할을 하고 있다고 말씀드릴 수 있겠습니다.

(사)한국인공지능학회 2021년 하계/추계 학술대회 논문심사위원장 김광수

코로나 19 판데믹이 지속되는 어려운 상황 속에서도 온라인으로 진행된 2021년도 한국인공지능학회 하계 학술대회가 성공적으로 잘 마무리가 되어서 감사하게 생각합니다.

모두가 직접 만나지는 못했지만, 각자의 자리에서 연구에 매진하고 계신 모습에서 어렵지만 그래도 희망을 가질 수 있었습니다.

특히, 5개의 최신 AI 분야에 대한 튜토리얼(김건희 교수님, 오민환 교수님, 이종욱 교수님, 김원화 교수님, 서민준 교수님), 세계 최고 석학을 모신 강화학습과 최적화에 관한 2개의 기조 강연 (Sergey Levine 교수님, Stephen Boyd 교수님), 그리고 AI와 데이터사이언스에 대한 깊은 통찰을 배울 수 있었던 2개의 초청 강연 (차상균 교수님, 정재승 교수님)을 진행하였는데 발표를 흔쾌히 수락해주신 연사들께 깊은 감사의 말씀을 올립니다. 또한 AI의 다양한 분야에 관한 7개의 기획 세션 발표자들과 LG AI 연구원과 ETRI에서 진행한 특별 세션 발표자들, 2개의 우수논문 세션 발표자들, 그리고 AI 대학원 설명회를 진행해주신 서울대/중앙대 AI 대학원장님들께 학회 프로그램을 꽉 채워주신 데 대해서도 진심으로 감사드립니다. 그리고, LG, 네이버, 한국중부발전, Vuno, KT, ETRI 등 저희 학술대회를 후원해주신 후원사에도 다시 한 번 감사 인사를 드립니다.

지난 2020년에 이어서 인공지능에 깊은 관심을 가지신 150여명께서 등록해주셨고, 저녁 7시까지 진행된 마지막 세션까지 100여명의 접속자들의 열띤 참여가 있어서 오프라인 학회 못지않은 열기를 느낄 수 있었습니다.

이제 반년이 지나서 논문집을 발간하고자 합니다. 여전히 다양한 분야의 인공지능 연구를 하고 계실 한국의 인공지능 연구자들의 노고와 열정에 감사하고, 향후 한국인공지능학회가 계속해서 한국의 인공지능 연구 및 응용에 중추적인 마중물 역할을 할 수 있기를 기대해봅니다.

2021년 하계학술대회 조직위원장 문태섭

2021 하계 및 추계학술대회 논문집을 발간하며

2020년에 이어, 2021년 또한 코로나 19라는 제약으로 인해 우리 모두에게 힘든 한 해가 되었습니다. 그러한 어려움 가운데에서도, 2021년 11월 4일부터 11월 6일까지 LG AI연구원과의 공동 개최를 통해 열린 한국인공지능학회 추계 학술대회는 한국 인공지능 기술의 발전에 있어서 큰 의미를 가지는 행사였습니다. 이번 추계 학술대회는 2개의 기조 강연, 1개의 특별강연, 6개의 튜토리얼, 8개의 기획 세션으로 구성되었습니다. 인공지능 분야의 원천 기술, 그리고, 핵심 응용 분야인 컴퓨터 비전, 자연어 처리 관련 연구를 비롯하여 수많은 논문들이 제출되었고, 국내 최고 수준의 연구자들이 해당 논문들을 심도있게 심사하고 의견을 나누며, 수준 높은 논문들을 엄선하여, 전례없는 수준의 학회 프로그램을 구성할 수 있었습니다. 이렇게 뜻깊은 학회를 열 수 있도록 해주신 많은 연구자분들과 심사위원분들에게 깊이 감사드리며, 특히 본 학술대회를 공동 개최하고 물심양면으로 후원해주신 LG AI연구원의 배경훈 원장님을 비롯한 관계자 여러분들에게 심심한 감사 인사를 드립니다. 이번 추계 학술대회를 통해, 긴밀한 산학 협력이 보다 활성화되어, 우리나라 인공지능 기술 발전의 초석이 될 수 있기를 희망합니다.

2021년 추계학술대회 공동조직위원장 배경훈, 주재걸

▣ 2021 한국인공지능학회 하계학술대회 조직위원회

• **회장**	유창동 교수 (KAIST)		
• **조직위원장**	문태섭 교수 (서울대)		
• **프로그램 위원장**	김광수 (KAIST)		
• **프로그램 위원**	이종욱(성균관대)	백승렬(UNIST)	정희철(경북대)
	김준영(KAIST)	석흥일(고려대)	장길진(경북대)
	문일철(KAIST)	권준석(중앙대)	서병기(UNIST)
	이주호(KAIST)	최영근(숙명여대)	김원화(POSTECH)
	주재걸(KAIST)	오민환(서울대)	박은병(성균관대)

▣ 2022 한국인공지능학회 | LG AI 연구원 추계학술대회

• **회장**	유창동 교수 (KAIST)		
• **조직위원장**	배경훈 원장 (LG AI 연구원)	주재걸 교수 (KAIST)	
• **학술위원회 위원장**	이승환 (LG AI 연구원)	김광수 교수 (KAIST)	
• **학술위원**	황성주(KAIST)	김준모(KAIST)	문태섭(서울대)
	최재식(KAIST)	신진우(KAIST)	양은호(KAIST)
	문일철(KAIST)	윤성로(서울대)	
•논문심사위원	이주호(KAIST)	장길진(경북대)	박은병(성균관대)
	문일철(KAIST)		
•운영위원	안소영/김유철/이용섭 (LG AI 연구원)		정두아(KAIST)

▣ 2021 하계학술대회 승인논문 목록

1. Hongjoon Ahn*[1], Jihwan Kwak*[1], Subin Lim*[1], Hyeonsu Bang*[1], Hyojun Kim*[1] and Taesup[2], SS-IL: Separated Softmax for Incremental Learning (Sungkyunkwan U., Seoul National U.)
2. Sungmoon Ahn and Shiho Kim, Analysis of Key-Features affecting traffic congestion prediction using Graph Neural Networks (Yonsei U.)
3. Sungmin Cha*[1], Beomyoung Kim*[2], Youngjoon[2] and Taesup Moon[1] SSUL: Semantic Segmentation with Unknown Label for Exemplar-based Class-Incremental Learning (Seoul National U., Naver clova)
4. Sungmin Cha[1], Naeun Ko[2], Youngjoon Yoo[2] and Taesup Moon[1], Self-Supervised Iterative Contextual Smoothing for Efficient Adversarial Defense against Gray- and Black-Box Attack (Seoul National U., Naver clova)
5. Jaehyeop Choi and Heechul Jung, SalfMix: A Novel Single Image-based Data Augmentation Technique using A Saliency Map (Kyungpook National U.)
6. Young-Geun Choi[1], Gi-Soo Kim[2], Seunghoon Paik[3] and Myunghee Cho Paik[3], Semi-Parametric Contextual Bandits with Graph-Laplacian Regularization (Sookmyung Women's U., UNIST, Seoul National U.)
7. Kang Haeyong and Yoo Chang D., Maximum Margin Loss Function for Imbalanced Dataset Learning (KAIST)
8. Seungwoo Im[1], Seokhun Jeon[2] and Hyoungkyu Song[1], Feature Selection with Normalized MIV Algorithm for a Blood Pressure Estimation Model (Sejoung U., Korea Electronic Technology Institute)
9. JoonHo Jang[1], Byeonghu Na[1], Dong Hyeok Shin[1], Mingi Ji[1], Kyungwoo Song[2] and Il-Chul Moon[1] , Unknown-Aware Domain Adversarial Learning for Open-Set Domain Adaptation (KAIST, U. of Seoul)
10. Jeongwoo Ju[1], Heechul Jung[1], Yoonju Oh[1] and Junmo Kim[1], Extending Contrastive Learning to Unsupervised Coreset Selection (KAIST, Kyungpook National U.)
11. Eunji Jun, Seungwoo Jeong, Da-Woon Heo and Heung-Il Suk, Parameter-efficient Multi-view Representation Learning for 3D Brain MRI (Korea U.)
12. Dong Un Kang[1], Dongwon Park[2] and Se Young Chun[1], Blur More To Deblur Better: Multi-Blur2Deblur For Efficient Video Deblurring (Seoul National U., UNIST)
13. Dahyun Kim, Sunjae Yoon, Ji Woo Hong and Chang D. Yoo, Semantic Association Network for Video Corpus Moment Retrieval (KAIST)
14. Dongjun Kim[1], Seungjae Shin[1], Kyungwoo Song[2], Wanmo Kang[1] and Il-Chul Moon[1], Score Matching Model for Unbounded Data Score (KAIST, U. of Seoul)
15. Doyeon Kim, Jooho Kim, Yechan Song, Jaeeun Park and Youngkeun Kim, Design of Reinforcement Learning Algorithm for Self-driving RC Car using Unity ML-Agent (Handong Global U.)
16. Jihu Kim and Kyusik Moon, Reducing End-to-end Delay in a Multithreaded Object Detection Application (Seoul National U.)

17. Junsu Kim, Younggyo seo and Jinwoo Shin, Training with Efficient Exploration in Goal-conditioned Hierarchical Reinforcement Learning (KAIST)
18. Junyeong Kim and Chang D. Yoo, Cause-and-Effect BART for Visual Commonsense Generation
19. Yoon-Yeong Kim[1], Kyungwoo Song[2], JoonHo Jang[1] and Il-Chul Moon[1], LADA: Look-Ahead Data Acquisition via Augmentation for Deep Active Learning (KAIST, U. of Seoul)
20. Oh Kwanseok, Yoon Jee Seok and Suk Heung-Il, Visual Explanation by Counterfactual Reasoning to a Classifier's Decision (Korea U.)
21. Jongmin Lee, Chanwoo Park and Ernest Ryu, A Geometric Structure of Acceleration and Its Rolein Making Gradients Small Fast (Seoul National U.)
22. Pilhyeon Lee[1], Jinglu Wang[2], Yan[2] and Hyeran[1], Separating Background by Uncertainty Estimation for Weakly Supervised Temporal Action Localization (Yonsei U., Microsoft Research Asia)
23. YongKyung Oh and Sungil Kim, Multi-channel Convolution Neural Network for Gas Mixture Classification (UNIST)
24. Kisu Ok, Inference Engine and Application based on Qualcomm Application Processor (Seoul National U.)
25. Chanwoo Park, Jisun Park and Ernest Ryu, Factor-$\sqrt{2}$ Acceleration of Accelerated Gradient Methods (Seoul National U.)
26. Jongjin Park, Sukmin Yun, Jongheon Jeong and Jinwoo Shin, Mitigating Class-distribution Mismatch for Semi-supervised Learning (KAIST)
27. Sungho Park[1], Sunhee Hwang[2], Dohyung Kim[1] and Hyeran Byun[1], FD-VAE: Fairness-aware Disentangling Variational Auto-Encoder for Representation Learning (Yonsei U., LG Uplus)
28. Trung Pham, Rusty John Lloyd Mina, Dias Issa and Chang D. Yoo, Self-supervised Learning with Local Attention-Aware Feature (KAIST)
29. Shulei Ren and Kyungsook Han, SVM Model for Predicting Lymph Node Metastasis in Individual Cancer Patients Based on miRNA-mediated RNA Interactions (Inha U.)
30. Muhammad Saqlain, Junuk Cha, Hansol Lee and Seungryul Baek, A Survey on Group Activity Recognition Techniques (UNIST)
31. Seungjae Shin[1], Byeonghu Na[1], Heesun Bae[1], Joonho Jang[1], Hyemi Kim[1], Kyungwoo Song[2] and Il-Chul Moon[1], Ordered Risk and Confidence Regularization for Robust Training from Biased Dataset (KAIST, U. of Seoul)
32. Byeong-Hoon So, Wook-Shin Han and Hyeonji Kim, A Semantic Equivalence Check Tool for Evaluating Text-to-SQL Models (POSTECH)
33. Junghyo Sohn, Eunjin Jeon , Wonsik Jung , Eunsong Kang and Heung-Il Suk, Residual Fine-Grained Attention: Tensor Calibrating to Elaborately Localize An Overall Object (Korea U.)
34. Ye-Chan Song, Do-Yeon Kim, Joo-Ho Kim, Jae-Eun Park Park and Young-Keun Kim, Implementation of Proximal Policy Optimization Algorithm for Autonomous Driving using AWS DeepRacer (Handong Global U.)
35. Sunjae Yoon and Chang D. Yoo, Cascaded Moment Proposal Network for Video Corpus Moment Retrieval (KAIST)

▣ 2021추계학술대회 승인논문 목록

1. Jaesin Ahn and Heechul Jung, SLIT: Shared Layers for Image Transformer (Kyungpook National U.)
2. Sangmin Bae, Sungnyun Kim and Se-Young Yun, Self-Contrastive Learning (KAIST)
3. Yoosung Bae, Yoonjeong Choi, Backdong Cha and Jeha Ryu, Gait Event Detection using Deep Learning (GIST)
4. Jae Soon Baik, Jun Ho Lee and Jun Won Choi, Long-Tailed Visual Recognition Using Bilateral Mixup Strategy (Hamyang U.)
5. Seong Jin Cho, Gwangsu Kim and Chang D. Yoo, Hypothesis Perturbation Strategy for Active Learning (KAIST)
6. Chaeyeon Chung[1], Taewoo Kim[1], Hyelin Nam[2], Seunghwan Choi[1], Gyojung Gu[1], Sunghyun Park[1] and Jaegul Choo[1], HairFIT: Pose-invariant Hairstyle Transfer via Flow-based Hair Alignment and Semantic-region-aware Inpainting (KAIST, Chung-Ang U.)
7. Sunhee Hwang, Minsong Ki, Seung-Hyun Lee and Sanghoon Park, Single CNN-based Fall Detection by Cut and Paste Data Augmentation (LG Uplus)
8. Joonhyun Jeong*[1], Sungmin Cha*[2], Youngjoon Yoo[1], Sangdoo Yun[1], Taesup Moon[2] and Jongwon Choi[3], Observations on K-image Expansion of Image-Mixing Augmentation for Classification (NAVER, Seoul National U., Chun-Ang U.)
9. Seungwoo Jeong, Wonjun Ko, Ahmad Wisnu Mulyadi and Heung-Il Suk, Continuous Riemannian Geometric Learning with Diffeomorphism Cholesky Mapping (Korea U.)
10. Donggyu Joo, Doyeon Kim, Hyounguk Shon and Junmo Kim, A Gift from Pre-Trained Network: Filter Pruning via Intermediate Model Recycling (KAIST)
11. Kyungmin Jo*, Gyumin Shim*, Sanghun Jung, Soyoung Yang and Jaegul Choo, CG-NeRF: Conditional Generative Neural Radiance Fields (KAIST)
12. Sungpil Kho1, Pilhyeon Lee1, Wonyoung Lee1, Minsong Ki2 and Hyeran Byun1, Improving Weakly Supervised Semantic Segmentation by Alleviating Texture Bias of CNNs (Yonsei U., LG Uplus)
13. Jaehyung Kim1, Dongyeop Kang2, Sungsoo Ahn3 and Jinwoo Shin1, What Makes Better Augmentation Strategies? Augment Difficult but Not too Different (KAIST, U. of Minnesota, Mohamed bin Zaeyed U. of AI)
14. Eunji Lee1, Sundong Kim2, Sihyun Kim1, Sungwon Park1, Meeyoung Cha2, Soyeon Jung3, Suyoung Yang3, Yeonsoo Choi3, Sungdae Ji3, Minsoo Song3 and Heeja Kim3, Classification of Goods Using Text Descriptions With Sentences Retrieval (KAIST, IBS, Korea Customs Service)
15. Jinhee Kim1, Taesung Kim1, Taewoo Kim1, Yoon-Ji Kim2, Dong-Wook Kim3, Byungduk Ahn4, In-Seok Song5 and Jaegul Choo1, Morphology-Aware Interactive Keypoint Estimation In Cephalometric X-Ray Images (KAIST, U. of Ulsan College of Medicine, Korea U. Anam Hospital Priavate, Korea U. Anam Hospital)
16. Sookyoung Kim and Hyejeong Jeon, A Causal Graph Learning and Exaplainable AI-based Root Cause Analysis Scheme for Home Appliance Products (LG Electronics)
17. Junghyun Lee1, Gwangsu Kim1, Matt Olfat2, Mark Hasegawa-Johnson3 and Chang D. Yoo1, MMD-based Fair PCA via Manifold Optimization (KAIST, UC Berkely/Citadel, UIUC)

18. Jungsoo Lee*1, Jooyeol Yun*1, Sunghyun Park1, Yonggyu Kim2 and Jaegul Choo1, Improving Face Recognition with Large Age Gaps by Learning to Distinguish Children (KAIST, Korea U.)
19. Hyesu Lim, Jimin Hong, Taehee Kim and Jaegul Choo, AVocaDo: Strategy for Adapting Vocabulary to Downstream Domain (KAIST)
20. Jihoon Tack1, Sihyun Yu1, Jongheon Jeong1, Minseon Kim1, Sung Ju Hwang1,2 and Jinwoo Shin1, Consistency Regularization for Adversarial Robustness (KAIST, AITRICS)
21. Taesung Kim1, Jinhee Kim1, Yunwon Tae2, Cheonbok Park3, Jang-Ho Choi4 and Jaegul Choo1, Reversible Instance Normalization for Accurate Time-Series Forecasting against Distribution Shift (KAIST, VUNO, NAVER, ETRI)
22. Chanwoo Kim1, Hoseong Cho2, Hansol Lee2, Yungseong Cho3 and Seungryul Baek3, Survey on Graph Attention Networks for Computer Vision Research (Kyungpook National U., Inha U., UNIST)
23. Jihyeon Kim, Changhwa Lee, Seongyeong Lee, Junuk Cha, Hansoo Park, Donguk Kim and Seungryul Baek, Finger Bendedness Classification from 3D Pose Regression (UNIST)
24. Taesung Kim1, Jinhee Kim1, Wonho Yang2, Hunjoo Lee3 and Jaegul Choo1, Missing Value Imputation of Time-Series Air-Quality Data via Deep Neural Networks (KAIST, Daegu Catholic U., CHEM. I. NET, Ltd.)
25. Youngho Kim and Je Hansoo Park, Yunyoung Jeong, Yunhoe Ku, Seungeun Lee and Seungryul Baek, A Survey on Recent 3D Hand Pose Datasets (UNIST)
26. Youngho Kim and Jeha Ryu ha Ryu, Physics-Based Regularization for Data-efficient and Robust State Transition Model Learning (GIST)
27. Myeonghun Lee and Kyoungmin Min, Molecular Graph-based Conditional Variable Autoencoder for De Novo Drug Design (Soongsil U.)
28. Sangwon Lee1, Hyemi Kim2 and Gil-Jin Jang1, Asymmetric U-Net for Weakly Supervised Sound Event Detection (Kyungpook National U., ETRI)
29. Yeongtak Oh1,2 and Changhee Han2, IDFAS: Informative Dual-Feature Aggregation Scheme for Continual Learning (Seoul National U., Korea Military Academy)
30. Sungho Park, Jewook Lee, Pilhyeon Lee and Hyeran Byun. Improving Fairness through Fair Contrastive Learning (Yonsei U.)
31. Jaeun Phyo, Wonjun Ko, Eunjin Jeon and Heung-Il Suk, Learning Deep Neural Network for Automatic Sleep Staging with Auxiliary Tasks for Effective Contextual Encoding (Korea U.)
32. Eojindl Yi, Juyoung Yang and Junmo Kim, Enhanced Prototypical Learning for Unsupe rvised Domain Adaptation in LiDAR Semantic Segmentation (KAIST)
33. Sunjae Yoon, Dahyun Kim and Chang D. Yoo, Cascaded Moment Proposal Network for Video Corpus Moment Retrieval (KAIST)

▣ 수록 논문 목차

◈ 하계

◈ 추계

한국인공지능학회 2021 하계 및 추계 학술대회 논문집

하계학술대회 논문

A Survey on Group Activity Recognition Techniques

Muhammad Saqlain[1], Junuk Cha[1], Hansol Lee[1] and Seungryul Baek[1]

[1] Graduate School of Artificial Intelligence, Ulsan National Institute of Science and Technlology (UNIST), Ulsan, Republic of Korea, {saqlain, jucha, hansollee, srbaek}@ unist.ac.kr

Abstract

Group activity recognition is an important problem in video understanding and has many practical applications. To understand the scene of multiple persons, the model needs to not only describe the individual action of each actor in the context but also infer their collective activity. This paper focuses on providing an overview of the recent advances in the field of group activity recognition. Additionally, it also discusses the commonly used datasets for group activity recognition.

Keywords— *Computer Vision, Group Activity Recognition, Deep Learning*

I. Introduction

Human activity recognition is one of the main areas of research in computer vision where the goal is to classify what a human is doing in an input video or image. Group activity recognition is a subset of human activity recognition problem which focuses on the collective behavior of a group of people, resulted from the individual actions of the persons and their interactions [1]. Collective activity recognition is a basic task for automatic human behavior analysis in many areas like surveillance, social behaviour understanding, or sports videos analysis.

In group activity recognition, it is crucial to take into account the individual actions of the people and their interactions as in many cases the group activity is formed by these actions and interactions. So, most group activity recognition models analyzes the personal activities either explicitly or implicitly. Some works try to predict individual actions and group activity in a joint framework using probabilistic graphical models [4] or neural networks that implement the functionality of graphical models [6]. Other approaches, use pooling methods like max pooling [9] or attention pooling [14] on the individual person representations to model the relation between individual humans and the collective activity.

Another important factor in recognizing the group activity recognition is the temporal development of the individual actions and group activity. Various approaches use Recurrent Neural Networks (RNNs) to model individual actions and group activity over time [21, 14]. Some existing works try to inject temporal data through Convolutional Neural Networks (CNNs) on optical flow fields calculated between two consecutive frames as an additional input to each time step of RNN [21, 11]. Recently, multi-stream convolutional networks where temporal information is modeled by CNNs on optical flow fields outperformed RNNs in action recognition task [20].

Due to the recent traction in the surveillance and security issues, the goal of this paper is to provide a closer look into human action recognition both as individual and group action recognition. In addition, it would also be necessary to discuss datasets which have mostly been used by these approaches. It would also be insightful to determine how these datasets would scale well when dealing with real-world anomaly detection.

II. Human Action Detection

We categorize action detection works into two parts like single person activity recognition and group activity recognition.

A. Single Person Activity Recognition

There have been numerous literature on human activity recognition. We can categorize them into various groups by the input cues such as RGB cue, depth cue, and pose/skeleton cue, for which each algorithm is using.

RGB is the most trivial inputs as it can be trivially obtained from the camera. Also, there have been numerous deep learning architectures such as VGG-16 [16] and ResNet [7] which could be utilized as the spatio-temporal feature extractors via their pre-trained weights. One of the traditional way of encoding spatio-temporal information using RGB streams is via using LSTM [12] on the CNN features. 3DConvNets [18] have also been popular way of encoding spatio-temporal information from RGBs. More systematic way of combining flow and RGB cues with longer range were investigated in [2].

The interest point detection and description has been widely studied [15] to provide reliable features for de-

scribing humans, objects, or scenes. The spatio-temporal interest points (STIPs) were often adopted [10] for compact representations of activities and events. Recent efforts [24, 25], therefore, have been devoted to developing reliable interest points and tracks for depth sequences. The interest points are extracted from low-level pixels [3] or mid-level parts [27]. In contrast to using local points, a holistic representation [26] was recently popular as it has shown generally effective and computationally efficient. Wang et al. [22] defined Hierarchical Dynamic Motion Maps (HDMM) by using different offsets between frames and extracting CNN features from them. More recently, Rahmaniet al. proposed a view-invariant descriptor HOPC [13] to deal with the 3D action recognition from unknown and unseen views. The use of skeleton joints has been suggested by [19] for alleviating ambiguities in action recognition.

B. *Group Activity Recognition*

There have been efforts to use probabilistic graphical models to tackle the group activity recognition problems. In [17], authors used a latent graph model for multi-target tracking, activity group localization, and group activity recognition. Initial probabilistic approaches used handcrafted features as input to their model. With the recent success of deep neural networks in various computer vision tasks, these networks were incorporated in the probabilistic group activity recognition models as feature extractors and inference engines. In [1], authors proposed a multistream convolutional framework for the task of group activity recognition in which new input modalities were easily incorporated into the model by the addition of new convolutional streams. Another work explored the idea of using RNNs for message passing [6]. They also proposed gating functions for learning structure of the graph.

Ibrahim et al. proposed a novel deep architecture that modeled group activities in a principled structured temporal framework [9]. Their 2-stage approach modeled individual person activities in its first stage, and then combined person-level information to represent group activities. The model's temporal representation was based on the LSTM. In [8], authors presented a hierarchical relational network that computes relational representations of people, given graph structures describing potential interactions. Relational representations of each person were created based on their connections in the particular graph. Their approach can learn relational feature representations that can effectively discriminate person and group activity classes. Recently, Wu et al. proposed a flexible and efficient Actor Relation Graph (ARG) to simultaneously capture the appearance and position relation between actors/players [23]. The connections in ARG were automatically learned using the Graph Convolutional Network (GCN), and the inference on ARG were efficiently performed with standard matrix operations. The ARG was able to capture the discriminative relation information for group activity recognition and got stat-of-the-art performance on the standard datasets.

III. Existing Benchmark Datasets

This section discusses in detail the publicly available datasets for the task of group activity recognition. There are a few papers which have created their own datasets but most of the works have tried to at least use one benchmark dataset in order to evaluate the performance of their proposed approaches with respect to previously published works. A summary presenting a high-level view of two benchmark datasets included in this subsection can be seen in Table 1.

A. *Volleyball dataset*

The Volleyball dataset [9] is composed of 4830 clips gathered from 55 volleyball games. Each clip is labeled with one of 8 group activity labels (right set, right spike, right pass, right winpoint, left set, left spike, left pass and left winpoint). The middle frame of each clip is annotated with the players' bounding boxes and their individual actions from 9 personal action labels (waiting, setting, digging, failing, spiking, blocking, jumping, moving and standing).

B. *Collective Activity dataset*

The Collective Activity dataset [5] contains 44 short video sequences (about 2500 frames) from 5 group activities (crossing, waiting, queueing, walking and talking) and 6 individual actions (NA, crossing, waiting, queueing, walking and talking). The group activity label for a frame is defined by the activity in which most people participate.

Dataset	Frames	Group/Individual	Ref.
Volleyball	4830	8/9	[9]
Collective Activity	2500	5/6	[5]

Table 1. Overview of Group Activity Recognition Datasets.

IV. Conclusion

This paper has provided an overview of the recent advances in group activity recognition methods. In addition, this paper has also presented two benchmark datasets along with important details. Over time, it can be seen that the datasets are gradually increasing in size and are also becoming closer to real-life scenarios. However, there is still an issue of manually annotating these videos. Approaches that leverage weakly-supervised learning should be explored more in the hopes that it might be able to automatically annotate videos.

REFERENCES

[1] Sina Mokhtarzadeh Azar, Mina Ghadimi Atigh, and Ahmad Nickabadi. A multi-stream convolutional neural network framework for group activity recognition. *arXiv preprint arXiv:1812.10328*, 2018.

[2] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[3] Zhongwei Cheng, Lei Qin, Yituo Ye, Qingming Huang, and Qi Tian. Human daily action analysis with multi-view and color-depth data. In *European Conference on Computer Vision*, pages 52–61. Springer, 2012.

[4] Wongun Choi and Silvio Savarese. A unified framework for multi-target tracking and collective activity recognition. In *European Conference on Computer Vision*, pages 215–230. Springer, 2012.

[5] Wongun Choi, Khuram Shahid, and Silvio Savarese. What are they doing?: Collective activity classification using spatio-temporal relationship among people. In *2009 IEEE 12th international conference on computer vision workshops, ICCV Workshops*, pages 1282–1289. IEEE, 2009.

[6] Zhiwei Deng, Arash Vahdat, Hexiang Hu, and Greg Mori. Structure inference machines: Recurrent neural networks for analyzing relations in group activity recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4772–4781, 2016.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[8] Mostafa S Ibrahim and Greg Mori. Hierarchical relational networks for group activity recognition and retrieval. In *Proceedings of the European conference on computer vision (ECCV)*, pages 721–736, 2018.

[9] Mostafa S Ibrahim, Srikanth Muralidharan, Zhiwei Deng, Arash Vahdat, and Greg Mori. A hierarchical deep temporal model for group activity recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1971–1980, 2016.

[10] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[11] Xin Li and Mooi Choo Chuah. Sbgar: Semantics based group activity recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 2876–2885, 2017.

[12] Asier Mujika, Florian Meier, and Angelika Steger. Fast-slow recurrent neural networks. *arXiv preprint arXiv:1705.08639*, 2017.

[13] Hossein Rahmani, Arif Mahmood, Du Huynh, and Ajmal Mian. Histogram of oriented principal components for cross-view action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 38(12):2430–2443, 2016.

[14] Vignesh Ramanathan, Jonathan Huang, Sami Abu-El-Haija, Alexander Gorban, Kevin Murphy, and Li Fei-Fei. Detecting events and key actors in multi-person videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3043–3053, 2016.

[15] Radu Bogdan Rusu, Gary Bradski, Romain Thibaux, and John Hsu. Fast 3d recognition and pose using the viewpoint feature histogram. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2155–2162. IEEE, 2010.

[16] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[17] Lei Sun, Haizhou Ai, and Shihong Lao. Localizing activity groups in videos. *Computer Vision and Image Understanding*, 144:144–154, 2016.

[18] Gül Varol, Ivan Laptev, and Cordelia Schmid. Long-term temporal convolutions for action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1510–1517, 2017.

[19] Jiang Wang, Zicheng Liu, Ying Wu, and Junsong Yuan. Learning actionlet ensemble for 3d human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 36(5):914–927, 2013.

[20] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pages 20–36. Springer, 2016.

[21] Minsi Wang, Bingbing Ni, and Xiaokang Yang. Recurrent modeling of interaction context for collective activity recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3048–3056, 2017.

[22] Pichao Wang, Wanqing Li, Zhimin Gao, Jing Zhang, Chang Tang, and Philip O Ogunbona. Action recognition from depth maps using deep convolutional neural networks. *IEEE Transactions on Human-Machine Systems*, 46(4):498–509, 2015.

[23] Jianchao Wu, Limin Wang, Li Wang, Jie Guo, and Gangshan Wu. Learning actor relation graphs for group activity recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9964–9974, 2019.

[24] Lu Xia and JK Aggarwal. Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2834–2841, 2013.

[25] Xiaodong Yang and YingLi Tian. Super normal vector for activity recognition using depth sequences. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 804–811, 2014.

[26] Xiaodong Yang, Chenyang Zhang, and YingLi Tian. Recognizing actions using depth motion maps-based histograms of oriented gradients. In *Proceedings of the 20th ACM international conference on Multimedia*, pages 1057–1060, 2012.

[27] Yang Zhou, Bingbing Ni, Richang Hong, Meng Wang, and Qi Tian. Interaction part mining: A mid-level approach for fine-grained action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3323–3331, 2015.

Implementation of Proximal Policy Optimization Algorithm for Autonomous Driving using AWS DeepRacer

Do-Yeon Kim[1], Joo-Ho Kim[1], Ye-Chan Song[1], Jae-Eun Park[1] and Young-Keun Kim*[1]

[1] School of Mechanical and Control Engineering, Handong Global University, Pohang. {21600053,21600144,21600369}@handong.edu

Abstract

This paper is focused on studying reinforcement learning for autonomous driving and implementing it on a small-scaled racing car. This paper designed and trained a PPO(Proximal Policy Optimization) algorithm for the autonomous driving in the simulation environment and deployed the trained model on the AWS DeepRacer to driving around the closed track. This paper explains how the reward function was designed for the optimal driving and how the hyperprameters were selected. The model was trained for four hours in the simulation environment provided by AWS. The designed reinforcement learning algorithm was validated by testing the trained model on the DeepRacer to circulate in the indoor track.

Keywords— *Reinforcement Learning, Deep Learning, Machine Learning*

I. Introduction

Developing self-driving system by using reinforcement learning has gained a high attention recently. Since it is difficult to train the agent in the real world environment, the reinforcement learning model is usually trained in a simulation environment using toolkits such as OpenAI gym and AWS DeepRacer simulator [1].

These toolkits allows the user create a custom environment of a simple track for a mini car. On these simulation environment, reinforcement algorithms can be developed and evaluated to modify the model for the optimal performance.

However, there exists some difficulties on deploying trained model on the actual world environment. Some essential input data such as precise location and heading direction may not be available in the actual world as they are in the simulation.

AWS DeepRacer is a recent system that has a high integration between the simulation environment and the real world for the autonomous driving of a mini car. It provides an online service to train and evaluate the reinforcement learning models in a simple simulated environment. The tranied model can be easily deployed on the DeepRacer vehicle of 1/18th scale RC car.

As a study on reinforcement learning for autonomous driving, this paper designed and trained PPO(Proximal Policy Optimization) algorithm [4] on the DeepRacer console and deployed on the vehicle for validation. This paper describes how the reward function and hyperparameters are designed and modified based on the learning rate and progress rate performance. Also, the experiment setup in the real-world track is expalined and the outcome is analyzed to validate the proposed algorithm.

II. Design Process of PPO Algorithm

The Proximal Policy Optimization (PPO) algorithm is a policy gradient method for reinforcement learning that optimally update based on the two networks of Actor and Critic. It is similar to well-known algorithms such as Actor-Critic, A2C, and A3C. However, PPO re-uses the a batch of experience data even after newly updating the policy. PPO leads to a less variance in training and helps the agent deviating to meaningless actions.

PPO aims to find the optimal parameter θ that maximizes the expected rewards by updating θ to minimize the cost function by using Trust Region method. It uses the ratio between the new policy(p_θ) and old policy($p_{\theta_{old}}$).

$$\theta \leftarrow \theta + \nabla_\theta \sum_{i=t-N+1}^{t} \frac{p_\theta(s_t,a_t)}{p_{\theta_{old}}(s_t,a_t)} \tag{1}$$

Instead of using a complex KL constraint, as in TRPO [2], PPO uses the policy ratio $r(\theta)$=(p_θ)/$(p_{\theta_{old}})$ to stay within a small interval ε around 1.

The objective function J_i of PPO that is maximized through the policy update is expressed as

$$J_i = \sum_{i=t-N+1}^{t} \frac{p_\theta(s_t,a_t)}{p_{\theta_{old}}(s_t,a_t)} A_i \tag{2}$$

where advantage A_i is the difference between the action value function $Q(a_i|s_i)$ and the state value function $V(s_i)$

for the action a_i at the state s_i. The positive advantage means the action taken by the agent is preferable. Since it is difficult to obtain the action value function directly, PPO estimates the advantage using generalized advantage estimation (GAE) [3].

$$A_i \triangleq Q(a_i|s_i) - V(s_i) \approx \sum_{k=i}^{t} (\gamma\lambda)^{k-i}\delta_k \quad (3)$$

where γ and λ are the GAE parameters and δ_k is the temporal-difference(TD) error defined by the reward R and state value function V.

$$\delta_k = R_{k+1} + \gamma V(s_{k+1}) - V(s_k) \quad (4)$$

This paper applied the clipped surrogate objective J_i^{clip} that saturates the advantage at a certain value and is known to have a better experimental performance than other surrogate objective functions.

$$J_i^{clip} \triangleq \min[ratio_i, A_i, clip(r_i 1-\varepsilon, 1+\varepsilon)A_i] \quad (5)$$

The update algorithm for PPO is summarized as shown in Figure 1.

Algorithm PPO

0. *Initialize* θ, w
Repeat 1~4
  1. *Collect N Sample* $(sample: \{s_i, a_i, s_{i+1})$
  Repeat 2~3$(Epoch)$
    2. *Actor update*: $\theta \leftarrow \theta + \alpha\nabla_\theta \sum_{i=t-N+1}^{t} J_i^{clip}$
    3. *Critic update*: $w \leftarrow w - \beta\nabla_\theta \sum_{i=t-N+1}^{t} \left(A_i^{GAE}\right)^2$
  4. *Clear the batch*

Fig. 1. PPO Update Algorithm

III. SIMULATION

A. Reward Function

The simulation environment of AWS console provides the vehicle data of the 2D position (x_{car}, y_{car}), heading angle (H), and steering angle (θ_s). The reward function from these parameter values is designed as shown in Fig 2.

The key idea is giving a positive reward if the steering angle of the wheels (θ_s) is similar to the net heading angle (θ_{net}) towards the target point from the current position of the vehicle. The target points are set as the points on the center line of the track at a certain distance away. If the steering angle (θ_s) is exactly the same as the net heading angle (θ_{net}), then the vehicle would drive directly to the target point.

B. Hyperparameters

The hyperparameters for training are set up as shown in Figure 3.The batch size was 64, not a large size, and the

Reward Function

Initializing & Updating Paramters
  H = *car heading angle*
  θ_s = *steering angle of car*
  x_t, x_y = *target point*
  x_{car}, y_{car} = *car point*
  eps = *error disired by the user*
  $dx = x_t - x_{car}$, $dy = y_t - y_{car}$
  $\theta_t = polar(dx, dy)$
  $\theta_{best\ angle} = \theta_t - H$

reward
  if $(\theta_{best\ angle} - \theta_s < eps)$
    get *reward*

Fig. 2. Overview of the Reward Function

Hyperparameter	Value
Gradient descent batch size	64
Entropy	0.01
Discount factor	0.95
Loss type	Huber
Learning rate	0.0003
Number of experience episodes between each policy-updating iteration	20
Number of epochs	10

Fig. 3. Hyperparamerters

epoch count was 10 to speed up the learning. In addition, the discount factor was set to 0.95, giving a large weight to the present state. It used Huber loss that is a compromised loss function between MSE and MAE.

C. Simulation Track

Fig. 4. Simulation Program

The training in the AWS console took four hours and the mean reward is shown to increases with the iterations, as shown in Figure 5. Furthermore, the average percentage

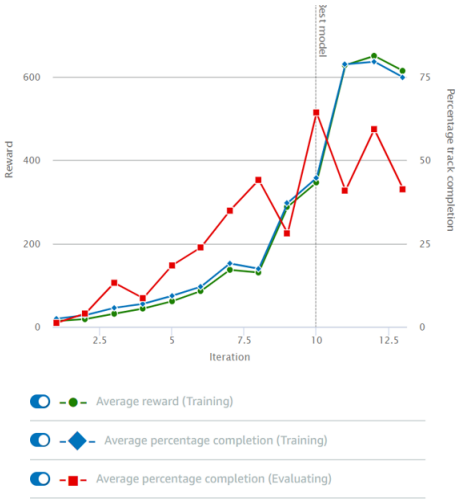


Fig. 5. Result Graph of Simulation

completion of training has also seemed to be increasing continuously.

IV. Real-World Implementation

The trained model was deployed on the DeepRacer vehicle with the specification as shown in Table 1

Table 1. Specs of Deep Racer

Car	18th scale of real car
CPU	Intel atom Processor
Memory	4GB RAM
Storage	32GB Memory
Wi-Fi	802.11ac
Camera	4 MP Camera with MJPEG
Softwatre	Ubuntu OS 20.04.3

Fig. 6. Real Track

The DeepRacer vehicle was tested on a small scale RC car track, as shown in Figure 6. Although it was a different environment from the simulation track, the vehicle managed to drive well without deviating off from the lanes. With the four hour trained model, the vehicle could be driven both the straight and curved lanes without any lane departure.

Fig. 7. DeepRacer driving within straight and curved lanes

V. Conclusion

This paper studied PPO of reinforcement learning to implement an autonomous driving RC car using the AWS DeepRacer vehicle. The reward function was designed to give a positive reward when the wheel steering angle is similar to the net heading angle of the vehicle towards the target point. The PPO model was trained in the simulation environment of AWS DeepRacer console and deployed on the vehicle. The driving test on the actual track showed the vehicle was able to drive within both the straight and curved lanes without any departure. The model could be improved with more training and modifying the designed simple reward function.

References

[1] Bharathan Balaji, Sunil Mallya, Sahika Genc, Saurabh Gupta, Leo Dirac, Vineet Khare, Gourav Roy, Tao Sun, Yunzhe Tao, Brian Townsend, Eddie Calleja, Sunil Muralidhara, and Dhanasekar Karuppasamy. Deepracer: Educational autonomous racing platform for experimentation with sim2real reinforcement learning. *CoRR*, abs/1911.01562, 2019.

[2] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.

[3] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

[4] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

A Semantic Equivalence Check Tool for Evaluating Text-to-SQL Models

Byeong-Hoon So[1], Hyeonji Kim[1] and Wook-Shin Han[†1]

[1] POSTECH, Computer Science and Engineering, {bhso, hjkim, wshan}@dblab.postech.ac.kr

Abstract

Text-to-SQL aims to enhance the accessibility of relational database management system for non-experts, building a model which translates a natural language text to SQL queries. Although a large number of recent researches, however, the accuracy measures they used are misleading; semantic equivalence of SQL queries has to be used to correctly measure the accuracy. Because existing semantic equivalence check tools support limited forms of SQL queries, we propose a new semantic equivalence check tool for the evaluation of text-to-SQL models. The proposed tool supports all operations required to support SQL queries that text-to-SQL models can generate. Experiments on eleven text-to-SQL datasets show the proposed tool successfully measures semantic accuracy with smaller measurement errors than that of state-of-the-art tools and previous accuracy measures.

Keywords— *Text-to-SQL, NL2SQL, NLIDB, Semantic parsing, Evaluation metric*

I. INTRODUCTION

Given a relational database and a natural language question to the database, translating natural language text to SQL query (text-to-SQL) is to find an SQL query to answer the question. This problem is important for enhancing the accessibility of relational database management systems for non-expert end-users. Accordingly, researchers have built a diversity of datasets [19, 43, 39] and improved model performances [26, 23, 29, 34, 2, 33, 17, 5, 32].

Nevertheless, all accuracy measures used in previous text-to-SQL researches are misleading. The following three automatic measures have been used: 1) result matching, 2) string matching, and 3) parse tree matching. Result matching executes the translated SQL query on one database instance and compares its results with that of the ground truth SQL query. It might create false positives: two equivalent queries could be labeled as inequivalent. False positives occur when two different queries return the same result by chance. In contrast, string matching and parse tree matching might create false negatives: two inequivalent queries could be labeled as equivalent. False negatives occur when two semantically equivalent queries are different in word-by-word or in parse trees.

The problem of determining whether two SQL queries are semantically equivalent is a long-standing problem in database research community. Accordingly, researchers have developed automated tools which search for proof that two SQL queries are semantically equivalent or inequivalent. To prove the equivalence of two SQL queries, they transform SQL queries into other semantic representations such as K-relations [16], UniNomial [9], and U-semiring [7] and search for an equivalence proof. Another approach [9] compiles SQL queries into constraints and find a counterexample by solving these constraints. Relying on formal methods, these tools reliably reason about equivalence for a restricted set of query types. However, they support limited forms of SQL queries because the semantic representations they used cannot support all operations in SQL syntax. For example, these representations cannot express sort operations and float comparisons, which are commonly appeared in text-to-SQL datasets.

In this paper, we propose a semantic equivalence check tool for SQL queries to measure the semantic accuracy of a text-to-SQL. We extend the multi-level validation tool in [21] by analyzing the ratio of query pairs filtered by each component and arranging the components in order with the lowest failure rate. The proposed tool is designed to support all operations required to support SQL queries that text-to-SQL models can generate. To this end, instead of transforming SQL queries into other semantic representations from other research community, the proposed tool uses query rewriting techniques and test data generation techniques from the database research community. Also, the proposed tool reliably reason the semantic equivalence, so it never determine equivalent query pairs as inequivalent or inequivalent query pairs as equivalent. Instead, the proposed tool may return unknown if it fails to prove both equivalence and inequivalence.

In this paper, we evaluate and analyze the proposed tool

This work was supported by Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No. 2018-0-01398, Development of a Conversational, Self-tuning DBMS)

†: corresponding author

using SQL queries in eleven text-to-SQL datasets. Furthermore, we compare the previous accuracy measures and the proposed tool and analyze the equivalent and inequivalent query pair separately to show the usefulness of the proposed tool. For the ground truth SQL queries in eleven text-to-SQL datasets and translated queries from one text-to-SQL model, the proposed tool resolves 97.39% and 96.53% of equivalent and inequivalent SQL query pairs, respectively. Specifically, the proposed tool correctly fix 10.08%p of wrong determination which result matching caused and 3.97%p of wrong determination which parse tree matching caused.

In summary, the main contributions of this paper are as follows:

- We show that all accuracy measures in the previous studies are misleading.
- We propose a semantic equivalence check tool that has sufficient coverage to measure the semantic accuracy of text-to-SQL models for the first time.
- An in-depth evaluation and analysis using eleven text-to-SQL datasets.

The remainder of the paper is organized as follows. Section ii. contains related work. Section iii. gives the motivations for a new semantic equivalence check tool for text-to-SQL models. Section iv. describes the architecture of the proposed semantic equivalence check tool. Section vi. shows the evaluation results. Section vi. concludes the paper.

II. Related work

A. Translating natural language text to SQL query

Translating natural language text to SQL query is a long-standing problem in both database community and natural language community which called the construction of natural language interfaces for databases [21]. Recently, methods based on deep learning [19, 43, 33, 22, 3, 35, 18, 44, 37, 38, 5, 17] have been actively proposed in the NLP community by exploiting state-of-the-art deep learning technologies. One of the main challenges in developing a text-to-SQL model with deep learning is the lack of training data. Accordingly, a new text-to-SQL benchmarks are published [43, 39]. WikiSQL [43] and Spider [39] benchmarks measure accuracy using parse tree matching. Since SQL queries of WikiSQL are very limited in their form, parse tree matching does not generate any problem. However, accuracy measurement using parse tree matching on the Spider benchmark is not accurately measured.

B. Semantic equivalence check tools

Previous studies on the semantic equivalence of SQL queries have been focused on applying formal methods. They developed semantic equivalence check tools which reliably reason about semantic equivalence for limited forms of SQL queries. These tools transform SQL queries into other semantic representations such as K-relations [16], UniNomial [9], and U-semiring [7]; then they search for an equivalence proof. For an inequivalence proof, Cosette [9] used a compiler that translates the input SQL queries $Q1$ and $Q2$ into constraints $C1$ and $C2$, Then, Cosette subsequently uses a constraint solver to find counterexamples by solving the formula $C1 \neq C2$. However, they support limited forms of SQL queries because the semantic representations they used cannot support all operations in SQL syntax. For example, these representations cannot express sort operations and float comparisons, which are commonly appeared in text-to-SQL datasets.

Recently, the multi-level validation tool is proposed for evaluating the semantic equivalence of SQL queries [21]. In this paper, we propose a semantic equivalence check tool based on this multi-level validation framework.

C. Test data generation for SQL queries

Software testing is an expensive component of software development and maintenance. Because RDBMSs strongly rely on SQL queries to to manage and manipulate their data, developers should properly test SQL queries. Although generating test data can be done by a human, the generation of data that test a SQL query becomes a difficult and time-consuming task as the SQL query becomes more complex. Accordingly, researchers have proposed approaches to automatically generate test data to support developers testing SQL queries [4, 12, 20, 30, 6].

Previous approaches [4, 12, 20, 30] transformed test data generation problem into a constraint satisfaction problem. These approaches defined mappings from the SQL language to constraints and solve the constraint satisfaction problem. Subsequently, they used constraint solvers to generate test data solving the constraints. However, because of high complexity of the mapping and limitations of solver tools, these approaches commonly get stuck in JOIN expressions, subqueries, date/time functions, and string manipulations.

Recently, EvoSQL [6] modeled the test data generation for SQL queries as a search-based problem. Search-based test data generation techniques have been applied in various software testing scenarios such as white-box unit testing [24] and regression testing [36]. Also, these techniques have been successfully generate complex data structures like Java objects [14] and string search problems [1]. Using these techniques, EvoSQL overcame the limitations caused by the high complexity of constraint satisfaction approaches.

III. Motivation

Previous text-to-SQL methods used result matching, string matching, or parse tree matching to measure accu-

Table 1. Failure cases of existing accuracy measures

Result matching	Query 1	**SELECT** name **FROM** *T* **WHERE** age > 20
	Query 2	**SELECT** name **FROM** *T*
String matching	Query 3	**SELECT** name **FROM** *T* **WHERE** age >= 20 **AND** age < 30
	Query 4	**SELECT** name **FROM** *T* **WHERE** age < 30 **AND** age >= 20
Parse tree matching	Query 5	**SELECT DISTINCT** *T1*.name **FROM** *T1* **WHERE** *T1*.id **IN** (**SELECT** id **FROM** *T2*)
	Query 6	**SELECT DISTINCT** *T1*.name **FROM** *T1* **JOIN** *T2* **ON** *T1*.id = *T2*.id

racy. Result matching executes the translated SQL query on the given database instance and compares its results with that of the ground truth SQL query. However, two different queries could return the same results by chance. For example, Query 1 and Query 2 in Table 1 returns the same results if all values in 'age' column are larger than 20. Thus, this measure might determine inequivalent query pairs as equivalent which leads to overestimating the accuracy. String matching compares two queries word-by-word. This measure might determine equivalent query pairs as inequivalent for various reasons, including the order of conditions, the projected columns, or joined tables. For example, Query 3 and Query 4 in Table 1 contain the same conditions in their WHERE clause, but the only difference is the order of conditions. However, the string match fails to know that these two queries are semantically equivalent. Parse tree matching compares parse trees from two SQL queries. This measure could prevent some errors in string matching, but it still determine equivalent query pairs as inequivalent. For example, Query 5 and Query 6 in Table 1 are a nested query and its flattened query. The two queries are semantically equivalent, but their parse trees are different from each other. In particular, the parse tree matching is an accuracy measure used in the leaderboards of WikiSQL [43] and Spider [39] datasets. So, measurement errors of this measure might mislead other studies.

Table 2. Comparison of accuracy measures (%) and measurement errors (%p).

Dataset	Accuracy (true value)	result matching	parse tree matching
ATIS	66.22	72.48 (+6.26)	7.61 (−58.61)
Advising (query split)	0.27	29.86 (+29.59)	0.27 (−0.00)
Advising (question split)	44.85	72.77 (+27.92)	44.15 (−0.70)
GeoQuery	69.64	78.21 (+8.57)	69.29 (−0.35)
Scholar	43.58	56.42 (+12.84)	37.61 (−5.97)
Patients	68.81	69.72 (+0.92)	68.81 (−0.00)
Restaurant	63.75	76.25 (+12.50)	63.75 (−0.00)
MAS	51.61	53.23 (+1.61)	46.77 (−4.84)
IMDB	16.67	19.05 (+2.38)	16.67 (−0.00)
YELP	0.00	29.27 (+29.27)	0.0 (−0.00)
WTQ	2.03	5.48 (+3.45)	1.83 (−0.20)

Table 2 shows large measurement errors of existing measures by comparing the accuracy on various text-to-SQL datasets. In this experiment, we measure the accuracy of the translated SQL queries by using result matching, string matching, and parse tree matching. We used NSP [19] for the text-to-SQL model. Then, we compared these values with the semantic accuracy: semantic equivalence between the translated SQL queries and the ground truth SQL queries. Note that the semantic accuracy ('semantic equivalence' in Table 2) represents the true value of accuracy and are verified manually. The values in parentheses indicate measurement errors (i.e. the difference between a measured value and its true value). The experiment results show that accuracy measured by result matching, string matching, and parse tree matching differ from semantic accuracy by up to 29.59% on Advising (query split), 66.22% on ATIS, and 58.61% on ATIS, respectively.

Despite a lot of research on semantic equivalence, the state-of-the-art tools such as Cosette [10, 8] support limited forms of SQL queries. Although this approach has the advantage of formally prove the equivalence of queries, there is a fundamental limitation that the semantic representation used in these tools cannot express some operations and database constraints.

To check whether the existing tool is usable for text-to-SQL evaluation, we tested the ratio of SQL queries supported by the state-of-the-art tool, Cosette. When testing with queries in eleven text-to-SQL datasets, Cosette supports 11.3% of queries because the semantic representations used in Cosette cannot express sort operations, float comparisons, HAVING clause, and expressions in GROUP BY clause. Furthermore, Cosette could not prove equivalence for all queries it supports because Cosette cannot fully support the constraints contained in the given database schema. For example, the foreign keys are not supported because they are difficult to model using Thus, we need a new semantic equivalence check tool that has sufficient coverage to measure the semantic accuracy of text-to-SQL models.

IV. Proposed Method

The proposed tool takes in a pair of SQL queries and a database instance and returns either "equivalent," "inequivalent," or "unknown." This tool is composed of three steps: run Cosette, result matching, and syntactic matching. Figure 1 shows the architecture of the proposed tool.

The first step of the proposed tool is 'run Cosette' (Figure 1 a.). We used Cosette to leverage the advance of existing research about semantic equivalence of SQL queries. Although this tool supports limited forms of SQL queries, it does not return wrong determination. Instead, the tool returns unknown (or timeout) if it fails to prove the equivalence and inequivalence. Thus, the proposed tool skips the other steps when Cosette proves the equivalence or inequivalence. If the queries are supported by Cosette or it returns unknown, the query pair is passed to the next step.

The second step of the proposed tool is 'syntactic matching' (Figure 1 c.). In this step, each query is rewritten into an equivalent normalized query. Then we compare

the parse trees of the normalized queries. The query pair is determined to "equivalent" if the parse trees of the normalized queries are the same. If the parse trees are different, the query pair is passed to the next step.

The last step of the proposed tool is 'result matching' (Figure 1 b.). This step executes the queries in various database instances. A query pair is determined to "inequivalent" if the queries return different execution results on any database instances. If the queries return the same execution results for all testing instances, the proposed tool fails to determine the semantic equivalence of two input queries and returns "unknown."

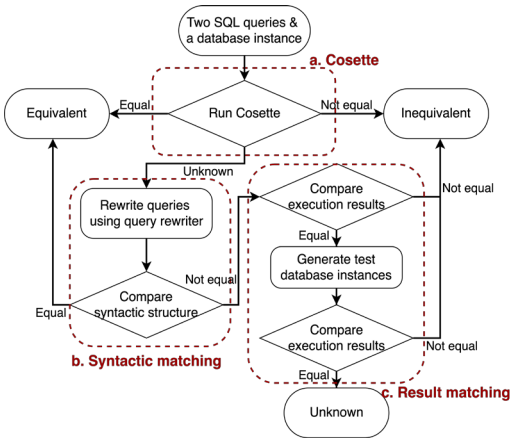


Fig. 1. A flowchart of the proposed semantic equivalence check tool.

A. *Run Cosette*

We opted Cosette for the first step of the proposed tool, because Cosette is the state-of-the-art equivalence check tool. Cosette takes two input queries and a database schema and returns equivalent, inequivalent, or unknown. Cosette returns equivalent when finding the formal proof and returns inequivalent when finding a counterexample which is a database instance that the input queries return different execution results. Thus, when Cosette returns equivalent or inequivalent, the proposed tool follows this answer and skips the other steps. When Cosette does not support the input queries or returns unknown, the input is passed to the next step.

B. *Syntactic matching*

In the second step, the proposed tool checks if the input queries are equivalent by comparing the syntactic structures of the queries. Previous text-to-SQL studies [43, 39] use parse tree matching to compare the syntactic structures of the query pairs. They compared the parse trees of the input queries and determined as equivalent only the parse trees are the same. However, as discussed in Section iii., the queries could be equivalent although their parse trees are different. To alleviate this problem, we normalized queries using query rewriting techniques before comparing the parse trees of them. Query rewriting is an automatic transformation that takes a query and database schema and yields a set of different queries, which are semantically equivalent to the input query. A normalized query refers to a query that can represent this set of queries. Therefore, if two queries are equivalent under query rewriting techniques, both queries will have the same normalized query.

For the implementation, we used the query rewriter in a commercial DBMS, IBM DB2, to transform an SQL query into a equivalent normalized query. Specifically, we used 'db2exfmt' command and get a normalized query in the explain table. After normalized the queries, the proposed tool uses a parse tree matching to determine whether the normalized query are semantically equivalent. That is, the tool determines that two queries are equivalent when every node in on parse tree has the corresponding matching node in the other. Otherwise, the input is passed to the next step.

C. *Result matching*

In the third step, the proposed tool checks if the input queries are inequivalent using result matching. Result matching was used in previous text-to-SQL studies [19, 39]. According to the definition of semantic equivalence, equivalent queries always return the same result when executed on any database instance. Obviously, it is impossible to compare execution results on all possible database instances, because there can be an infinite number of different database instances. Instead, if the input queries return different execution results on a database instance, we can prove that these queries are inequivalent using that instance as a couterexample. Since text-to-SQL datasets always includes a database instance for each query, we can use this instance for result matching. However, as semantically different queries happens to have the same execution results on the database instance, we generate more database instances for result matching. Especially, when the input queries return empty results, we cannot prove the queries are inequivalent even if they are completely different. To address this problem, we generate database instances where a query returns non-empty results.

We use a database instance generator to generates database instances for result matching. In database testing, we generate database instances, run queries over these instances, and find bugs in database engines. A database instance generator used in database testing could generate a database instance with non-empty results for an input query. Using this database instance generator, we generate different instances so that every query has non-empty results for at least one of these instances. Thus, we can effectively resolve that two queries are semantically inequivalent by executing on the generated instances. If the input queries return the same results for the given instance and all generated instances, the proposed tool fails to determine

the semantic equivalence of two input queries and returns "unknown."

For the implementation, we used EvoSQL [6] for the database instance generator. Given an SQL query and an empty database instance, EvoSQL searches for records that satisfy the query and inserts the records into the empty instance. We generated one instance for each query and compared the execution results of a query pair on three database instances, a given instance and two generated instances. If EvoSQL failed to generate the records for a query, we skipped generating an instance for that query.

V. EXPERIMENTS

A. *Datasets*

In our experiments, we use eleven text-to-SQL datasets: ATIS, Advising (query split), Advising(question split), GeoQuery, Scholar, Patients, Restaurants, MAS, IMDB, YELP, WTQ. Each text-to-SQL dataset covers different patterns of SQL queries. Therefore, it is important to use multiple datasets to cover various syntax and patterns of SQL queries. We categorized the datasets into three groups according to their characteristics. The followings are characteristics of each group.

The datasets in the first group has been widely used for semantic parsing, which is the task of translating an English question into a formal meaning representation. These datasets were used as text-to-SQL datasets by transforming the formal meaning representation into SQL. ATIS [28, 11, 19, 42], Advising [13], GeoQuery [40, 41, 27, 15, 19], Scholar [19], Patients [3], and Restaurants [31, 27] are in this group. All queries in each dataset are on a single domain such as flight booking, US geography, or restaurant booking. This dataset consists of real-world questions and SQL queries written by experts to answer those questions. Although SQL queries included in each dataset repeat a small number of patterns, but overall, these datasets have various SQL queries including join, nested queries, grouping, and ordering query.

The datasets in the second group has been proposed to evaluate text-to-SQL models. MAS, IMDB, and YELP are in this group. Microsoft Academic Search provides a database of academic social networks and a set of queries for MAS dataset and the authors of NaLIR [23] select the queries of them. The authors of SQLizer [34] provides IMDB and YELP datasets which are a movie database and a business reviewing database. These datasets is on a single domain like the first group of datasets. However, these datasets do not contain ground truth SQL queries, so we manually wrote an SQL query for each question. Fortunately, the questions are complex enough to include various SQL queries containing join, grouping, and nested queries.

The datasets in the last group are provided for question answering on tables. The WTQ dataset [21] is in this group. This dataset consists of 9,287 randomly sampled questions from the existing WikiTableQuestions [25]. The WikiTableQuestions consists of questions and answers for web tables on various domains. The SQL queries are not provided in WikiTableQuestions, so the queries are collected through crowd-sourcing. These SQL queries include various patterns of SQL queries such as domain specific functions and self join, but they only address single table queries.

Table 3 shows the statistics of the eleven datasets. For the experiments, we use the same data split (i.e., training, validation, and test examples) if it is provided by the authors of each dataset. Otherwise, we perform the random split with the ratio of 11:1:5.6, for training, validation, and test examples, respectively. For the Advising dataset, we use two versions of splits published in [13], namely question-based split and query-based split. The question-based split is a traditional method used in the other datasets. This method regards each (English question, SQL query) pair as a single item so that each pair belongs to either a training set, a validation set, or a test set. Meanwhile, the query-based split ensures that each SQL query belongs to either a training set, a validation set, or a test set. For ATIS, we manually removed 93 incorrect examples.

Table 3. The statistics for the text-to-SQL datasets.

Dataset	Total queries	Training queries	Validation queries	Test queries	Tables	Rows	Size (MB)
ATIS [28, 11, 19, 42]	5317	4379	491	447	25	162k	39.2
Advising [13] (query split)	4387	2040	515	1832	15	332k	43.8
Advising [13] (question split)	4387	3585	229	573	15	332k	43.8
GeoQuery [40, 41, 27, 15, 19]	880	550	50	280	7	937	0.14
Scholar [19]	816	498	100	218	10	144M	8776
Patients [3]	342	214	19	109	1	100	0.016
Restaurant [31, 27]	251	157	14	80	3	18.7k	3.05
MAS [23]	196	123	11	62	17	54.3M	4270
IMDB [34]	131	82	7	42	16	39.7M	1812
YELP [34]	128	80	7	41	7	4.48M	2232
WTQ [21]	9287	5804	528	2955	2102	58.0k	35.6

B. *Experimental design*

We designed experiments to measure the semantic accuracy of the text-to-SQL model using eleven text-to-SQL datasets described in Section A. and one text-to-SQL model. This experiments aim to show how accurate the proposed tool is and show that each component of the tool is useful.

In each experiment, we trained a text-to-SQL model for each text-to-SQL dataset. Then, the proposed tool takes a pair of SQL queries, a generated SQL query from the model and the corresponding ground truth SQL query, and a given database instance and checked the semantic equivalence of the queries. We measure the ratio at which the tool successfully determines the semantic equivalence of

these query pairs. Also, to show that each step of this tool is meaningful, we measured the ratio of resolved query pairs at each step. Although the possible answer is whether equivalent or inequivalent, the tool could answer "equivalent," "inequivalent," or "unknown." If the tool answers "unknown," the evaluation result is measured as failure, not correct or wrong answer. Finally, we measure the ratio of correct answer, wrong answer, and failure.

For the experiments, we choose NSP [19] as the text-to-SQL model because it supports all operations of SQL queries in the datasets. We trained the model and performed grid search using the hyper-parameters in Table 4. Then, the trained model generates a translated SQL query of test set. Table 5 shows the number of equivalent and inequivalent query pairs for each datasets.

Table 4. Hyper-parameters.

The dimension of a word embedding vector	{100, 300}	Learning rate (LR)	{1e-3, 1e-4}
The number of layers	{1, 2}	Batch size	{64, 200}
Dropout rate	{0.3, 0.5}	LR decay	{1, 0.98, 0.8}
The dimension of context vector		{50, 300, 600, 800}	

Table 5. The number of equivalent and inequivalent query pairs for each dataset.

Dataset	Total query pairs	Equivalent query pairs	Inequivalent query pairs
ATIS	447	296	151
Advising (query split)	1832	5	1827
Advising (question split)	573	257	316
GeoQuery	218	95	123
Scholar	218	95	123
Patients	109	75	34
Restaurants	80	51	29
MAS	62	32	30
IMDB	42	7	35
YELP	41	0	41
WTQ	2955	60	2895
Total	6639	1073	5566

C. *Comparison of accuracy measures*

Figure 6 shows the accuracy measured by the proposed tool and previous accuracy measures, and then compares measurement errors of them. Note that a query pair is marked as resolved when the proposed tool correctly label the query pair as "equivalent" or "inequivalent." And we say a query pair is unresolved when the proposed tool returns "unknown" for that query pair. Because of unresolved query pairs, the accuracy measured by the proposed tool is expressed as a range. The values in parentheses indicate the measurement errors of each accuracy measures. The measurement errors of result matching and parse tree matching are false positives and false negatives, respectively. We skip the string match because parse tree matching is always better than string match. The results show that the proposed tool correctly determines 10.08%p of total queries determined wrong by result matching and 3.97%p of total queries determined wrong by parse tree matching. Especially, the proposed tool reduces measurement errors of result matching and parse tree matching by up to 29.27%p (YELP) and 56.37%p (ATIS), respectively. The proposed tool also has the advantage that it is more convenient to handle errors in the future by labeled them as unknown instead of creating false positives or false negatives. These results show that the proposed tool can replace result match or parse tree match when evaluating text-to-SQL models.

D. *Resolved query pairs for each step*

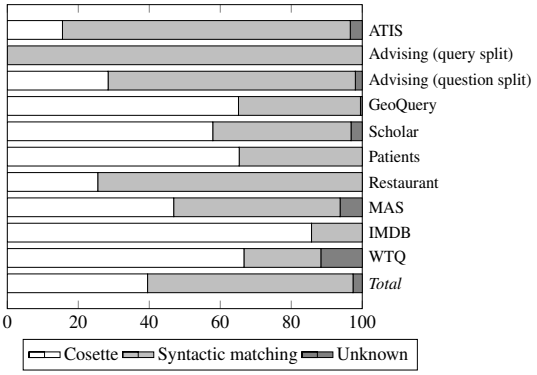


Fig. 2. Resolved query pairs (%) for each step when the queries in each pair are equivalent.

Figure 2 shows the ratio of semantically equivalent query pairs resolved at each step except result matching. We hide result matching in this figure because result matching does not resolve any equivalent query pairs. Comparing with the all equivalent query pairs, the proposed tool resolved 39.52% and 57.88% of query pairs in Cosette and syntactic matching, respectively, and fail to resolve the remaining 2.61% of query pairs. This means that the proposed tool showed an improvement of 57.88%p (from 39.52% to 97.39%) compared to Cosette. Especially, the proposed tool showed a significant performance improvement in 81.08%p, 100%p, and 74.51%p on ATIS, Advising (query split), and Restaurant datasets, respectively.

Figure 3 shows the ratio of inequivalent query pairs resolved at each step. We hide syntactic matching in this figure because syntactic matching does not resolve any inequivalent query pairs. And we divide the result matching into 'result matching on a given instance' and 'result matching on generated instances' depending on which instance returns different execution results. Comparing with the all inequivalent query pairs, the proposed tool resolved 5.70%, 80.09%, and 10.74% of query pairs in Cosette, result matching on the given instance, and result matching on generated instances, respectively, and fail to resolve the remaining 3.47% of query pairs. This means that the proposed tool showed the improvement of 90.83%p (from 5.70% to 96.53%) compared to Cosette. Result matching

Table 6. Comparison of accuracy measures and measurement errors.

Dataset	Semantic equivalence	Semantic equivalence check tool [Ours]	Result matching	Parse tree matching
ATIS	66.22	63.98-70.25 (6.27)	72.48 (+6.26)	7.61 (−58.61)
Advising (query split)	0.27	0.27-2.62 (2.35)	29.86 (+29.59)	0.27 (−0.00)
Advising (question split)	44.85	43.98-51.13 (7.15)	72.77 (+27.92)	44.15 (−0.70)
GeoQuery	69.64	69.29-74.29 (5.00)	78.21 (+8.57)	69.29 (−0.35)
Scholar	43.58	42.20-46.33 (4.13)	56.42 (+12.84)	37.61 (−5.97)
Patients	68.81	68.81-69.72 (0.92)	69.72 (+0.92)	68.81 (−0.00)
Restaurant	63.75	63.75-63.75 (0.00)	76.25 (+12.50)	63.75 (−0.00)
MAS	51.61	48.39-51.61 (3.23)	53.23 (+1.61)	46.77 (−4.84)
IMDB	16.67	16.67-16.67 (0.00)	19.05 (+2.38)	16.67 (−0.00)
YELP	0.00	0.00-0.00 (0.00)	29.27 (+29.27)	0.0 (−0.00)
WTQ	2.03	1.79-4.30 (2.51)	5.48 (+3.45)	1.83 (−0.20)
Total	16.16	15.74-18.93 (3.19)	29.85 (+13.69)	11.81 (−4.35)

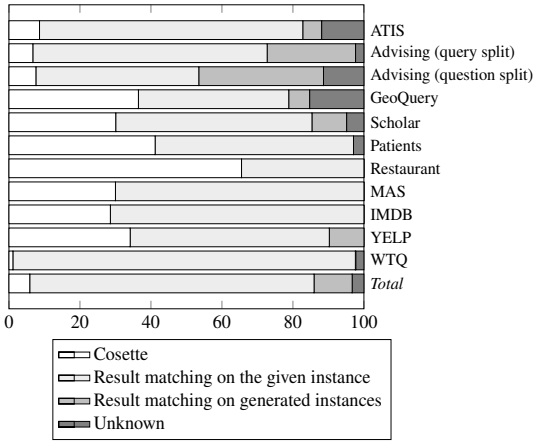


Fig. 3. Resolved query pairs (%) for each step when the queries in each pair are inequivalent.

on a given database failed to resolve 0% to 46.52% of inequivalent query pairs depending on the dataset, but this ratio decreased up to 15.29% when using generated instances.

E. Ablation study

Table 7. The unresolved query pairs (%) when skipping each component.

Dataset	*Equivalent query pairs*			*Inequivalent query pairs*		
	All steps	**w/o syntactic matching**	**w/o Cosette**	**All steps**	**w/o test data generation**	**w/o Cosette**
ATIS	3.38	84.46	3.38	11.92	17.22	11.92
Advising (query split)	0.00	100.0	0.00	2.35	27.26	2.35
Advising (question split)	1.95	71.60	1.95	11.39	46.52	11.71
GeoQuery	0.51	34.87	0.51	15.29	21.18	21.18
Scholar	3.16	42.11	3.16	4.88	14.63	10.57
Patients	0.00	34.67	0.00	2.94	2.94	2.94
Restaurants	0.00	74.51	0.00	31.03	31.03	34.48
MAS	6.25	53.13	6.25	0.00	0.00	3.33
IMDB	0.00	14.29	0.00	0.00	0.00	2.86
YELP	-	-	-	0.00	9.76	4.88
WTQ	11.67	33.33	11.67	2.31	2.42	3.42
Total	2.61	60.48	2.61	3.47	14.21	4.37

We tested performance of the proposed tool while skipping each component of the proposed tool to show that all components are necessary. Figure 7 shows the ratio of unresolved query pairs when skipping each component of the proposed tool. For the equivalent query pairs, we tested the ratio when skipping 1) syntactic matching and 2) Cosette. Note that there is no equivalent query pairs on YELP, so there are no values of it. The results show that 57.87% (from 2.61% to 60.48%) of equivalent query pairs could be resolved by syntactic matching. Surprisingly, if we only compared the equivalent query pairs, the ratio for unresolved pairs does not changed when we skip to run Cosette. This results mean that the rewriting rules of IBM DB2 cover the ability to determine equivalent query pairs in Cosette, at least in the datasets we used. For the inequivalent query pairs, we tested the ratio when skipping 1) result matching on test data generation and 2) Cosette. The results show that 10.74% (from 3.47% to 14.21%) of inequivalent query pairs could be resolved by result matching on generated database instances. Also, 0.90% (from 3.47% to 4.37%) of inequivalent query pairs could be resolved by Cosette. We find some cases when result matching on generated instances could fail when it is difficult to return different execution results for two queries with random records satisfying a query. In this case, only Cosette could resolve the query pairs because Cosette searches an instance where the execution results of the two queries are different. Here is a simplified example which only Cosette determine the inequivalence: "**SELECT COUNT**(1) **FROM** Mountain" and "**SELECT COUNT**(altitude**) FROM** Mountain". While the first query returns the number of rows in Mountain table, the second query returns the number of rows excluding the rows that the altitude column is null.

F. Failure case analysis

We analyze the failure cases (i.e. unresolved query pairs) of the proposed tool. Figure 4 shows sampled failure cases of inequivalent query pairs. Both Query 1 and Query 2 in Figure 4 (b) seems to return 'the most highest mountain." However, if there exists a tie in the target column of sort operation, Query 1 returns all mountains that are tied, whereas Query 2 returns only one of them. Figure 4 (a) is a counterexample that Query 1 and Query 2 are inequivalent. For this reason, Query 1 and Query 2

Mountain

name	altitude	state_name
Grays	4349	Colorado
Antero	4349	Colorado
Nebo	3636	Utah

State

state_name	population
Colorado	2889000
Utah	1461000

(a) A sample database instance.

		Execution result	
Query 1	**SELECT** name **FROM** Mountain **WHERE** altitude = (**SELECT MAX**(altitude) **FROM** Mountain);	→	name Grays Antero
Query 2	**SELECT** name **FROM** Mountain **ORDER BY** altitude **DESC LIMIT** 1;	→	name Grays

(b) The case with tie in max aggregation.

		Execution result	
Query 1	**SELECT** population **FROM** State **WHERE** state_name **IN** (**SELECT** state_name **FROM** Mountain **WHERE** altitude = 4349);	→	population 2889000
Query 2	**SELECT** State.population **FROM** State, Mountain **WHERE** State.state_name = Mountain.state **AND** Mountain.altitude = 4349;	→	population 2889000 2889000

(c) The case with join on non-unique column.

Fig. 4. Examples of failure cases when two queries are inequivalent.

are not resolved if the given database instance and generated database instance do not have tie records. Query 1 and Query 2 in Figure 4 (c) is another example that the proposed tool fails to resolve. Query 2 could be seem as a flattened query of Query 1, however, Figure 4 (a) is a counterexample that Query 3 and Query 4 are inequivalent. Query 1 and Query 2 returns differently when the left hand side of the IN operation (Mountain.state_name) that satisfies the right hand side (Colorado) is non-unique. These failure cases stem from the test data generation algorithm that generates a non-empty return instance for one given query. Because this algorithm takes only one query, it cannot take into account the condition that can be derived from the two queries. Thus execution results on generated instances are likely to be the same by chance. In the case of equivalent query pairs, the proposed tool could failed to resolve when two queries include function operations such as string functions and algebraic functions (e.g. ROUND, CEIL). Because query rewriter of IBM DB2 does not change function expressions, conditions cannot be determined to be equivalent except when two function expressions are exactly the same.

VI. Conclusion

This paper implemented a new semantic equivalence check tool for SQL queries. Especially, the tool aims to be measure the semantic accuracy of text-to-SQL models. On the top of the state-of-the-art semantic equivalent equivalence check tool, the proposed tool could determine the equivalence of the query pairs by comparing the normalized queries of them. Also, the proposed tool could determine the inequivalence of the query pairs by generating test database instances and comparing the execution results on them. When we evaluated the proposed tool using eleven text-to-SQL datasets and a text-to-SQL model, the proposed tool successively resolve 97.39% of equivalent query pairs and 96.53% of inequivalent query pairs without any wrong determination. We expect that the proposed tool will be used in future researches to measure the semantic accuracy of text-SQL models.

References

[1] Mohammad Alshraideh and Leonardo Bottaci. Search-based software test data generation for string data using program-specific search operators. *Software Testing, Verification and Reliability*, 16(3):175–203, 2006.

[2] Christopher Baik, H. V. Jagadish, and Yunyao Li. Bridging the semantic gap with SQL query logs in natural language interfaces to databases. In *ICDE*, pages 374–385, 2019.

[3] Fuat Basik, Benjamin Hättasch, Amir Ilkhechi, Arif Usta, Shekar Ramaswamy, Prasetya Utama, Nathaniel Weir, Carsten Binnig, and Ugur Çetintemel. Dbpal: A learned nl-interface for databases. In *SIGMOD*, pages 1765–1768, 2018.

[4] Carsten Binnig, Donald Kossmann, Eric Lo, and M Tamer Özsu. Qagen: generating query-aware test databases. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pages 341–352, 2007.

[5] Ben Bogin, Jonathan Berant, and Matt Gardner. Representing schema structure with graph neural networks for text-to-sql parsing. In *ACL*, pages 4560–4565, 2019.

[6] Jeroen Castelein, Maurício Aniche, Mozhan Soltani, Annibale Panichella, and Arie van Deursen. Search-based test data generation for sql queries. In *Proceedings of the 40th international conference on software engineering*, pages 1220–1230, 2018.

[7] Shumo Chu, Brendan Murphy, Jared Roesch, Alvin Cheung, and Dan Suciu. Axiomatic foundations and algorithms for deciding semantic equivalences of sql queries. *arXiv preprint arXiv:1802.02229*, 2018.

[8] Shumo Chu, Brendan Murphy, Jared Roesch, Alvin Cheung, and Dan Suciu. Axiomatic foundations and algorithms for deciding semantic equivalences of SQL queries. *PVLDB*, 11(11):1482–1495, 2018.

[9] Shumo Chu, Chenglong Wang, Konstantin Weitz, and Alvin Cheung. Cosette: An automated prover for sql. In *CIDR*, 2017.

[10] Shumo Chu, Chenglong Wang, Konstantin Weitz, and Alvin Cheung. Cosette: An automated prover for SQL. In *CIDR*, 2017.

[11] Deborah A. Dahl, Madeleine Bates, Michael Brown, William M. Fisher, Kate Hunicke-Smith, David S. Pallett, Christine Pao, Alexander I. Rudnicky, and Elizabeth Shriberg. Expanding the scope of the ATIS task: The ATIS-3 corpus. In *ARPA Human Language Technology Workshop*, 1994.

[12] Michael Emmi, Rupak Majumdar, and Koushik Sen. Dynamic test input generation for database applications. In *Proceedings of the 2007 international symposium on Software testing and analysis*, pages 151–162, 2007.

[13] Catherine Finegan-Dollak, Jonathan K. Kummerfeld, Li Zhang, Karthik Ramanathan, Sesh Sadasivam, Rui Zhang, and Dragomir R. Radev. Improving text-to-sql evaluation methodology. In *ACL*, pages 351–360, 2018.

[14] Gordon Fraser and Andrea Arcuri. Evosuite: automatic test suite generation for object-oriented software. In *Proceedings of the 19th ACM SIGSOFT symposium and the 13th European conference on Foundations of software engineering*, pages 416–419, 2011.

[15] Alessandra Giordani and Alessandro Moschitti. Translating questions to SQL queries with generative parsers discriminatively reranked. In *COLING*, pages 401–410, 2012.

[16] Todd J Green, Grigoris Karvounarakis, and Val Tannen. Provenance semirings. In *Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 31–40, 2007.

[17] Jiaqi Guo, Zecheng Zhan, Yan Gao, Yan Xiao, Jian-Guang Lou, Ting Liu, and Dongmei Zhang. Towards complex text-to-sql in cross-domain database with intermediate representation. In *ACL*, pages 4524–4535, 2019.

[18] Po-Sen Huang, Chenglong Wang, Rishabh Singh, Wen-tau Yih, and Xiaodong He. Natural language to structured query generation via meta-learning. In *NAACL-HLT*, pages 732–738, 2018.

[19] Srinivasan Iyer, Ioannis Konstas, Alvin Cheung, Jayant Krishnamurthy, and Luke Zettlemoyer. Learning a neural semantic parser from user feedback. In *ACL*, pages 963–973, 2017.

[20] Shadi Abdul Khalek, Bassem Elkarablieh, Yai O Laleye, and Sarfraz Khurshid. Query-aware test generation using a relational constraint solver. In *2008 23rd IEEE/ACM International Conference on Automated Software Engineering*, pages 238–247. IEEE, 2008.

[21] Hyeonji Kim, Byeong-Hoon So, Wook-Shin Han, and Hongrae Lee. Natural language to sql: Where are we today? *Proceedings of the VLDB Endowment*, 13(10):1737–1750, 2020.

[22] Mirella Lapata and Li Dong. Coarse-to-fine decoding for neural semantic parsing. In *ACL*, pages 731–742, 2018.

[23] Fei Li and H. V. Jagadish. Constructing an interactive natural language interface for relational databases. *PVLDB*, 8(1):73–84, 2014.

[24] Phil McMinn. Search-based software test data generation: a survey. *Software testing, Verification and reliability*, 14(2):105–156, 2004.

[25] Panupong Pasupat and Percy Liang. Compositional semantic parsing on semi-structured tables. In *ACL*, pages 1470–1480, 2015.

[26] Ana-Maria Popescu, Alex Armanasu, Oren Etzioni, David Ko, and Alexander Yates. Modern natural language interfaces to databases: Composing statistical parsing with semantic tractability. In *COLING*, 2004.

[27] Ana-Maria Popescu, Oren Etzioni, and Henry A. Kautz. Towards a theory of natural language interfaces to databases. In *IUI*, pages 149–157, 2003.

[28] P. J. Price. Evaluation of spoken language systems: the ATIS domain. In *DARPA Speech and Natural Language Workshop*, pages 91–95, 1990.

[29] Diptikalyan Saha, Avrilia Floratou, Karthik Sankaranarayanan, Umar Farooq Minhas, Ashish R. Mittal, and Fatma Özcan. ATHENA: an ontology-driven system for natural language querying over relational data stores. *PVLDB*, 9(12):1209–1220, 2016.

[30] María José Suárez-Cabal, Claudio de la Riva, Javier Tuya, and Raquel Blanco. Incremental test data generation for database queries. *Automated Software Engineering*, 24(4):719–755, 2017.

[31] Lappoon R Tang and Raymond J Mooney. Automated construction of database interfaces: Integrating statistical and relational learning for semantic parsing. In *EMNLP*, pages 133–141. Association for Computational Linguistics, 2000.

[32] Bailin Wang, Richard Shin, Xiaodong Liu, Oleksandr Polozov, and Matthew Richardson. Rat-sql: Relation-aware schema encoding and linking for text-to-sql parsers. *arXiv preprint arXiv:1911.04942*, 2019.

[33] Xiaojun Xu, Chang Liu, and Dawn Song. Sqlnet: Generating structured queries from natural language without reinforcement learning. *CoRR*, abs/1711.04436, 2017.

[34] Navid Yaghmazadeh, Yuepeng Wang, Isil Dillig, and Thomas Dillig. Sqlizer: query synthesis from natural language. *PACMPL*, 1(OOPSLA):63:1–63:26, 2017.

[35] Semih Yavuz, Izzeddin Gur, Yu Su, and Xifeng Yan. Dialsql: Dialogue based structured query generation. In *ACL*, pages 1339–1349, 2018.

[36] Shin Yoo and Mark Harman. Regression testing minimization, selection and prioritization: a survey. *Software testing, verification and reliability*, 22(2):67–120, 2012.

[37] Tao Yu, Zifan Li, Zilin Zhang, Rui Zhang, and Dragomir R. Radev. Typesql: Knowledge-based type-aware neural text-to-sql generation. In *NAACL-HLT*, pages 588–594, 2018.

[38] Tao Yu, Michihiro Yasunaga, Kai Yang, Rui Zhang, Dongxu Wang, Zifan Li, and Dragomir R. Radev. Syntaxsqlnet: Syntax tree networks for complex and cross-domain text-to-sql task. In *EMNLP*, pages 1653–1663, 2018.

[39] Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir R. Radev. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. In *EMNLP*, pages 3911–3921, 2018.

[40] John M. Zelle and Raymond J. Mooney. Learning to parse database queries using inductive logic programming. In *AAAI*, pages 1050–1055, 1996.

[41] Luke S. Zettlemoyer and Michael Collins. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *UAI*, pages 658–666, 2005.

[42] Luke S. Zettlemoyer and Michael Collins. Online learning of relaxed CCG grammars for parsing to logical form. In *EMNLP-CoNLL*, pages 678–687, 2007.

[43] Victor Zhong, Caiming Xiong, and Richard Socher. Seq2sql: Generating structured queries from natural language using reinforcement learning. *CoRR*, abs/1709.00103, 2017.

[44] Ming Zhou, Guihong Cao, Ting Liu, Nan Duan, Duyu Tang, Bing Qin, Xiaocheng Feng, Jianshu Ji, and Yibo Sun. Semantic parsing with syntax- and table-aware SQL generation. In *ACL*, pages 361–372, 2018.

한국인공지능학회 2021 하계 및 추계 학술대회 논문집

추계학술대회 논문

IDFAS: Informative Dual-Feature Aggregation Scheme for Continual Learning

Yeongtak Oh[1,2] and Changhee Han[2]

[1] Seoul National University, Seoul, Korea, oyt9306@gmail.com
[2] AI R&D center, Korea Military Academy, Seoul, Korea, chhan46@gmail.com

Abstract

Recently, continual learning, also known as incremental learning, receiving lots of attention to mitigate catastrophic forgetting while training the network subsequently with the following data streams that contain new tasks not visited before. However, the parasitic stability-plasticity dilemma, as a trade-off between the sustainability and the flexibility of the network still remains troublesome. Thus, in this paper, we propose a dual-memory structure-based continual learning, by exploiting the stable and the plastic submodules in one network. Experiments show that the proposed learning scheme could efficiently increase the feature extraction performance by combining with the existing backbones on each incremental training stage; by adopting the data augmentation, and aggregating the deep features considering the feature aggregation flow and the information flow. On CIFAR-100, our method showed superior performance than other existing methods, under the multi-class incremental settings with several different phases.

Keywords— *Continual learning, Visual self-attention, Feature aggregation flow, Information flow, Mixed-Label Data augmentation*

I. Introduction

Up to date, humans commonly have two important learning characteristics: efficient learning competence of the new information (e.g, skills, both structured and unstructured knowledge), and keeping the useful information learned from previous events (e.g. cherishable memory, important information to survive). Thus, to reveal the mechanisms of the brain, among many research fields, *Complementary Learning System* (CLS) theory [14, 21] is suggested. Following the CLS theory, in the brain, the mechanisms of the brain memory can be mainly divided by two different parts, hippocampus, and neocortex, which play the role of experience generalization and memorizing episodic-like events, respectively.

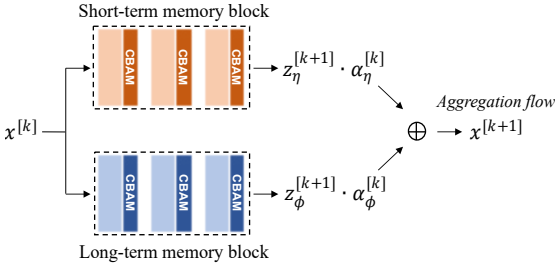


Fig. 1. Schematic illustration of the proposed learnable scheme implemented AANets, where $k \in [0,1,2]$, which denotes the level of the proposed architecture.

Recently, in the deep learning field, which achieved drastic successes in many fields(e.g. image recognition [6]), continual learning is now receiving lots of attention. However, *stability-plasticity* (SP) dilemma is still challenging to train the model to mitigate the catastrophic forgetting [13], and is a desiderata to achieve a long-lasting learning scheme. Especially, *Multi-class incremental learning* (MCIL) [12] is a well-known challenging subject to achieve the plausible performance of the model at the current task, by using a model trained on previous tasks via incrementally updating the model. Motivated from the current challenges of MCIL, we propose a new approach to overcome the existing limitations under the MCIL settings by adopting the feature aggregation flow and the information flow from the dual memory architecture to seamlessly train the network.

In this paper, we propose an efficient learning scheme named Informative Dual-Feature Aggregation Scheme for Continual Learning (IDFAS) that could be applicable to various existing strategies by considering the dual memory concepts with two blocks characterized by the plastic and the stable properties. Specifically, our proposed method exploits the convolutional self-attention module, data augmentation, feature aggregation flow and information flow.

II. Related Work

Typically, continual learning (CL) literature could be categorized by the problems they admire to solve. In this section, several approaches are introduced.

Modular-based approach renovates the model architecture by focusing on task-shared and task-adaptive param-

eter; in an adversarial manner [5], or a channel-wise task-specific gating module [1], and additive parameter decomposition [24], etc.

Manifold-based approach [4, 18] adopted the concept of manifold learning to incrementally learn the current task by utilizing the knowledge acquired from the previous tasks by considering *gradient trajectories* on the high-dimensional geometric space. By using the geometry of the manifold (e.g. Riemannian, Grassmann), these methods ease the problem of *intransigence* [4] while preserving knowledge learned from the previous tasks by implementing the gradual work on that space.

Regularization-based approach [3] adopts the strategy to restrict the important network parameters to retain the pre-trained knowledge from catastrophic forgetting. Although the regularization-based approach is advantageous than other approaches as it requires a small amount of network parameters, however, when the continual training phases of MCIL become long, such regularization could easily becomes obsolete, as it induces the *representation drift* [19].

Our paper is similar to the modular-based, and manifold-based approach, but a totally different thing from the above-mentioned previous approaches is that we consider a CL strategy with *aggregation-based approach* [11], which utilizes the dual memory-based structure to reflect on the duality of extracted features from each branch, not by from the conventional single model-based CL. In detail, we both exploited a feature aggregation flow and an information flow in a dual-memory architecture. Following the results of adaptive aggregation networks (AANets) [11], a single-memory-based learning strategy is limited to manipulate the weights to overcome the SP dilemma. Meanwhile, by adopting the dual-memory architecture, the desirable performance could be achievable. Thus, in this paper, our network's backbone was set as LUCIR [7] with a single dual-memory structure [11].

III. Our Proposed Framework

In the human recognition system, the hippocampal system enables the short-term memory to learn rapidly and to adaptively remember the episode experienced by surrounding environments. Conversely, the neocortex system enables the long-term memory to memorize and store important or impressive memory well. Thus, based on that concepts, we considered a dual memory structure as described in Figure 1. The divided feature extraction blocks are constituted with a long-term memory block and a short-term memory block at a specific level. For a long-term memory block, stability rather than plasticity is mainly preserved while network training. Besides, for a short-term memory block, plasticity rather than stability is mainly considered while network training. By doing so, plasticized information is extracted through the short-term memory block, and stabilized information is extracted through the long-term block. Further, two different represented features are aggregated and used as an input for the next-level feature extraction block in a network.

In a dual-memory system, as two feature maps extracted from each branch are simultaneously learned from the same input sources, we raised a doubt that *'In the MCIL settings, what is the effective learning method to guide the network to learn the important information by modulating and aggregating the two different extracted features (i.e., stability and plasticity)?'*. Thus, in this paper, we propose an effective and efficient learning scheme in four ways: 1) self-attention, named convolutional block attention module (CBAM) [22], 2) feature aggregation flow [11], 3) information flow [9], and 4) mixed-label data augmentation [25]. Note that, CBAM module is easily executable and contributes to the increment of the network performance; CBAM is mainly composed of sequentially arranged a channel attention module and a spatial attention module. By integrating two attention modules sequentially within a ResBlock, the spatial attention module plays the role of deciding 'where' to focus the channel-wise attention module exploits the inter-channel relationship. In figure 2, used CBAM module attached after each Resblock is described.

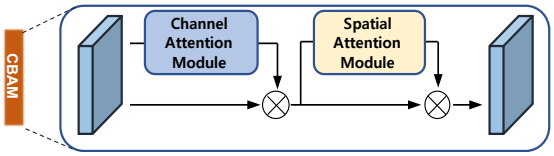


Fig. 2. Attached CBAM module architecture [22], where ⊗ denotes element-wise multiplication.

A. *Feature aggregation flow in a dual-memory system*

For better understating of our proposed method, following two notions are clarified as follows. *Multi-head vs Single-head*: as a single-head architecture is more challenging to achieve enhanced performance and more computationally efficient, we utilized a single-head based network structure. Next, *Feature aggregation vs Feature ensemble*: conventionally, feature ensemble is executed with multiple algorithms than exploiting one algorithm, to achieve enhanced predictive performance. In this paper, the feature aggregation is progressed inside of the feature extraction block, and flowed by passing sequential ResNet blocks. The feature extraction procedure was conducted with the set backbone model of 32-layer ResNet [6], and the learning rate for the stable and the plastic sub-modules are set as the same value.

For aggregation flow, the basic idea is very closely related to AANET [11]. Following the AANET baseline, the stable branch and the plastic branch constitute the whole feature aggregation modules. For stable blocks, its basic parameters in the convolutional layer are firstly learned at the zeroth-phase, and nearly froze in the following N subsequent phases. To this end, the structural patterns within the neuron are preserved and only the channel-wise masks

are slowly updated by the scaling and shifting weights. Detailed operations in the incremental neuron scaling are described as follows in eq. (1), where X_{i-1} means the input feature maps at the i^{th} layer. Conversely, for the plastic block, the neuron-level scaling is not considered, and the plastic block tries to learn the new tasks rather than retaining the previously learned knowledge.

$$X_i = (W_i \odot \phi_i) X_{i-1} \tag{1}$$

In detail, as described in Figure 1, feature aggregation flow [11] at k^{th} ResBlock is executed as follows in eq. (2).

$$\begin{aligned} x^{[k+1]} &= \alpha_{\phi}^{[k]} * x_{\phi}^{[k]} + \alpha_{\eta}^{[k]} * x_{\eta}^{[k]} \\ where, \alpha_{\phi}^{[k]} &+ \alpha_{\eta}^{[k]} = 1 \end{aligned} \tag{2}$$

Here, $x^{[k]}$ is the input of $k^{th}(\in [0,1,2])$ level ResBlock and k denotes the level of the proposed architecture. The constraint of eq. (2) could be accepted as the learnable scaling coefficients $\alpha_{\phi}^{[k]}, \alpha_{\eta}^{[k]}$, and due to that parameter, the network's weight could be adjusted well.

B. Information flow in two consecutive tasks

Next, for information flow, we focused on minimizing the inconsistency of the extracted features between the previous tasks and the current task. Thus, we propose the information flow to hold the gradual walk of the aggregated features by considering both the plasticity and the stability. To this end, we exploited the geodesic flow on the high dimensional manifold [18].

Following [18], the geodesic flow between two sequential tasks could be approximated with the pseudo inverse, by exploiting the singular value decomposition (SVD) [20] algorithm to get the orthogonal complement and the diagonal elements. Further, by projecting the intermediate subspaces, the mismatches of two embedding features could be minimized along the geodesic path. In detail, let the embedding features at task t and $t-1$ as z_t and z_{t-1}, the embedding features could be described with the models trained on the previous task and the current task, respectively. Thus, the definitions are as follows; $z_{t-1} = f(x, \Phi_{t-1})$, $z_t = f(x, \Phi_t)$, where f means the feature extractor and Φ is the model parameters. Further, the subspace P_{t-1} and P_t derived by using the principal component analysis (PCA), are the spaces spanned by the orthogonal basis of z_{t-1} and z_t, respectively. Refer to eq. (3), (top) original inner product between two consecutive extract features are (bottom) relaxed by considering the manifold structure along with the geodesic flow. Considering the geodesic flow, the inner product between two consecutive features is driven as a positive semi-definite matrix Q, which is a projected manifold structure into the intermediate subspace. The distillation loss considering the geodesic flow could be derived as eq. (4).

$$\begin{aligned} g_P(z_{t-1}, z_t) &= z_{t-1}^T P P^T z_t^T \\ g_{Geo}(z_{t-1}, z_t) &= z_t^T Q z_{t-1}^T \end{aligned} \tag{3}$$

$$\mathscr{L}_{Geo} = 1 - \frac{z_t Q z_{t-1}}{||Q^{1/2} z_t|| ||Q^{1/2} z_{t-1}||} \tag{4}$$

Thus, information flow could drastically improve the model performance by taking a bridge between the extracted features via 1) decomposing the behaviors of extracted features, 2) constructing a seamless geodesic path on the high-dimensional feature space.

For better understanding, our information flow could be interpreted as follows: alleviating the behavioral mismatches rather than re-arranging the mechanisms would show better generalization ability for CL under the MCIL setting. Where 'behavior' means the represented features of the outer feature extraction blocks, and 'mechanism' means the weights and the parameters of the model [9]. However, as a trade-off of the SP dilemma, the renovations on the information flow have two aspects: if the degree of the cosine similarities of the projected encoded features is too huge, the behaviors even be driven far away from the initial position, leading to the loss of informative knowledge from the previous tasks. Conversely, if the similarities are too small, the learning ability for new tasks could drastically be decreased. Thus, proper optimization of the geodesic flow would adequately minimize the catastrophic forgetting.

C. Mixed-Label Data Augmentation

Following [2, 15], it is revealed that implementing the simple CutMuix [25] algorithm could contribute to effectively enhance the network performance on MCIL settings. Thus, we exploited the CutMix algorithm to train the model by minimizing the categorical cross-entropy. In detail, detailed descriptions of the used CutMix algorithm are as follows; as the continual task iteration proceeds, due to the different exemplars, the selected samples from the current episodic memory follow the different data distribution from the previously learned data distributions. Thus, mixed-label data augmentation (ML-DA) could improve the feature extraction performance on the current learning phase by mixing the images in the classes of the new tasks and the classes of the old tasks, stored in the memory. To do so, ML-DA with CutMix algorithm could mitigate the mismatches of the current and the previous data distributions. Implemented CutMix algorithm from the selected samples to generate a mixed-sample and a smoothed label is as follows in eq. (5).

$$\begin{aligned} \tilde{x} &= \mathbf{m} \odot x_1 + (1 - \mathbf{m}) \odot x_2 \\ \tilde{y} &= \lambda y_1 + (1 - \lambda) y_2 \end{aligned} \tag{5}$$

where (x_1, y_1), (x_2, y_2) denotes the labeled samples, $\mathbf{m}$ means the randomly selected pixel regions of the input image x_1. Here, the random pixel regions are picked by following the beta-distribution determined by the hyperparameter β. Note that, following our experiments, the CutMix algorithm is effective when training on the large

training epochs. In our research, for training epochs, 250, and 120 epochs were set at the zeroth and incremental phases, respectively. Thus, such ML-DA could reduce an overfitting problem during a training procedure. As the effectiveness of the CutMix training on the zeroth-phase of the continual learning, refer to the supplementary material section A.

D. *Total loss settings*

In this section, the basic architecture for feature extraction is based on the [7]. To reveal the rationale of the loss settings for the proposed learning scheme, we describe the total loss. Here, we brought the used loss [7], as described in equation (6).

$$\mathscr{L}_{LUCIR} = \frac{1}{|N|} * \sum_{x \in N} (L_{ce}(\tilde{x}, \tilde{y}) + \lambda L_{dis}^{G}(x)) + \frac{1}{|N_o|} \sum_{x \in N_o} L_{mr}(x) \tag{6}$$

The classification loss L_{ce} is the basic categorical cross-entropy loss, distillation loss with cosine normalization on the high-dimensional sphere is L_{dis}^{G} with class imbalance weight λ, and the top-K sampled margin ranking loss is L_{mr}. Note that, by adapting the CutMix algorithm, original $L_{ce}(x, y)$ is substituted as $L_{ce}(\tilde{x}, \tilde{y})$ by eq. (5). Consequently, the total loss proposed in this paper for better enhanced learning scheme is as follows in eq. (7).

$$\mathscr{L}_{total} = \mathscr{L}_{LUCIR} + \mathscr{L}_{Geo} \tag{7}$$

IV. Experimental Validations

In this section, we examine the proposed method by evaluating the performance of MCIL setting of CIFAR-100 dataset. The CIFAR-100 dataset [8] is composed of several data with 100 classes, and the color images (500 for train, 100 for test) of each class are set with the same size of 32 × 32. Further, to reveal the results of the proposed method, we constructed following results to support the superiority. In this paper, our computing environment is built with a 16-Core AMD Ryzen 9 3950X CPU, a single GEFORCE RTX 3080 GPU, and 128Gb RAM for hardware; PyTorch 1.8.0, CUDA 11.1, and Python 3.8 for the environmental setting. Note, following our computing resources, the total training time to proceed with one specific multi-phase (e.g. 10 phase) required about 19h until termination.

Total pseudo-code for the implemented our learning scheme with the baseline network is explained in Algorithm 1. Here, the previous learning mechanisms are written in black color, and the modified algorithm by applying our proposed learning scheme is highlighted in blue color.

A. *Ablation studies*

In this subsection, the results of several baseline models applied with the proposed learnable scheme to further improve the performance of the model under MCIL conditions are shown. Firstly, w/o CBAM, w/o information flow are compared with CIFAR-100 datasets. Due to the lack of computational resources, we did not cover the Imagenet dataset to verify the effectiveness of the network. In our paper, the initial learning rates for the stable block, the plastic block, and the fusion variables used for the feature aggregation are set as $0.1, 0.1, 1e^{-8}$, respectively. And the scheduler for the learning rate is designed to perform after the middle and three-quarters of the entire set epochs.

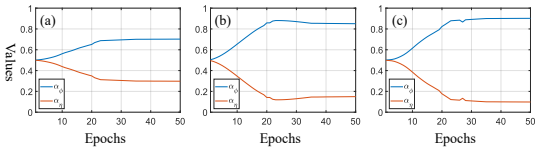


Fig. 3. The changes of the feature aggregation weights over 50 epochs in one representative training phase, where α_ϕ denotes the coefficient of the stable block, and α_η denotes the coefficient of the plastic block.

1) Feature aggregation flow: As proceeding the several ablation studies, we figured out one interesting one that attached the CBAM module induced the bigger variations of fusion variables during training. In detail, as we can see in Figure 3, at the first level, the changes of α_ϕ over the one training phase reached the specific value. In the following, the changes of α_ϕ in level 2 showed more drastic changes than level 1, and after fluctuation, two coefficients are reached to the specific value. Finally, at the last block, the changes of α_ϕ were not quite drastic and reached a stable point.

2) Information flow: Next, for the results of renovating the information flow, first, as we can see in the left and the middle plot of Figure 4, the decreasing slopes of the average and the cumulative accuracy were very similar. However, by following the results in the right plot in Figure 4, in terms of the decrement of the original accuracy, LUCIR with the proposed scheme showed the superior performance enhancement by overcoming the catastrophic forgetting. Consequently, by applying the information flow, the accuracy of the zeroth-phase is continually retained during the consecutive training phases. Thus, by considering the information flow, not only the total learning ability is greatly enhanced, but the catastrophic forgetting problem is significantly resolved through knowledge retention at the zeroth-phase. Thus, even the learning procedure terminates, the task-adapted model still could show the plausible performance by minimizing the forgetting.

Finally, total results are represented in Table 1 on two different multi-phase MCIL settings using with CIFAR-100 dataset. First, the previous approaches showed relatively poor performance in the MCIL settings. Conversely, as a result of the CutMix algorithm, overfitting is minimized and the model could achieve superior performance even for the long MCIL phases. Further, as we can see in the Table 1, on the MCIL setting, the performance of the comparative studies showed that in the case of knowledge adaptation, which means the ability to learn the new knowledge from the consecutive data streams, the

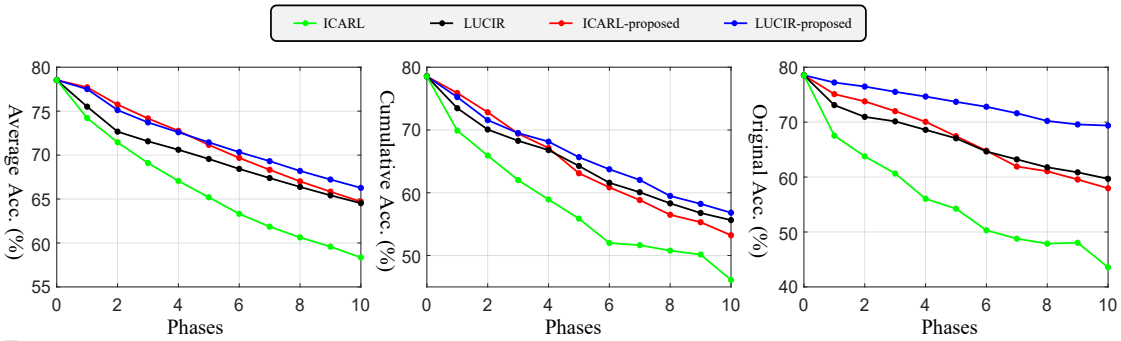


Fig. 4. Total accuracy plots (left: average, middle: cumulative, right: original) under the 10-Phase MCIL setting. By simply attaching the CBAM module, the slope performance decay became gentle. Further, by considering the information flow, the performance decay of the model drastically dwindled.

case of LUCIR as a baseline was better than the case of iCaRL when training on long MCIL phases. Furthermore, in terms of knowledge retention, which means the degree of retaining the pre-trained knowledge after the learning is terminated, LUCIR showed better performance than iCaRL. In addition, to verify the visualization results for the effects of the stability and the plasticity in each learning phase, refer to the supplementary material section B.

Algorithm 1: Pseudo code for the modified training procedure of AANET [11] with combining with the proposed learning scheme (in the i-th phase).

Input: New class data D_i; old class exemplars $\varepsilon_{0:i-1}$; old parameters $\alpha_{0:i-1}$, $\phi_{0:i-1}$, $\eta_{0:i-1}$; base model θ_{base}
Output:New parameters α_i, ϕ_i, η_i; new class exemplars ε_i
Get D_i and $\varepsilon_{0:i-1}$ from the memory
Initialize $[\phi_i, \eta_i]$ with $[\phi_{i-1}, \eta_{i-1}]$
Initialize α_i with α_{i-1}
Select exemplars ε_i on current task by herding [16]
for *epochs* **do**
  for *mini-batches* ***in*** $\varepsilon_{0:i-1} \cup D_i$ **do**
    Substitute $L_{ce}(x,y)$ with $L_{ce}(\tilde{x},\tilde{y})$ by Eq. 5.
    Compute Z_{t-1} and Z_t, and their projections P_{t-1} and P_t to get the information flow.
    Train $[\phi_i, \eta_i]$ on $\varepsilon_{0:i-1} \cup D_i$ by minimizing $\mathscr{L}_{LUCIR} + \mathscr{L}_{Geo}$.
  for *mini-batches* ***in*** $\varepsilon_{0:i-1} \cup \varepsilon_i$ **do**
    Substitute $L_{ce}(x,y)$ with $L_{ce}(\tilde{x},\tilde{y})$ by Eq. 5.
    Optimize α_i on $\varepsilon_{0:i-1} \cup \varepsilon_i$ by minimizing $L_{ce}(\tilde{x},\tilde{y})$.
  Update exemplars ε_i by herding
  Replace $\varepsilon_{0:i-1}$ with $\varepsilon_{0:i-1} \cup \varepsilon_i$

V. Conclusion

To sum up, our proposed learning scheme achieved drastically enhanced performance and alleviated the catastrophic forgetting in MCIL settings. Following several ablation studies, we showed that the proposed learning scheme achieved SOTA performance than the existing several benchmarks. In this paper, the proposed learning scheme is achieved by simply exploiting 1) the self-attention module right after the feature extraction modules, 2) the feature aggregation flow from the plastic and the stable blocks, 3) the information flow on the high-dimensional embedded feature space, and 4) the ML-DA to mitigate the overfitting during training. For future work, although our methods are empirically verified, but not theoretically developed, so we will build up the theoretical bounds to highlight the superiority. Further, some experiments will be proceeded with Imagenet-sub and Imagenet datasets to verify the robustness of our proposed learning scheme.

Table 1. Average incremental accuracies (%) of several CL strategies under 5, 10-Phase MCIL settings for CIFAR-100 dataset. In the 0-th phase, the initial model is trained on 50 classes, the remaining classes are given evenly in the subsequent phases.

Used methods	CIFAR-100	
Average Accuracy (%)	N = 5	N = 10
LwF [10]	49.59	46.98
BiC [23]	59.36	52.66
Mnemonics [12]	63.34	62.28
iCaRL [16]	57.12	52.56
iCaRL w/ CBAM	60.88	58.35
iCaRL w/ CBAM w/ Info	**67.52**	**64.69**
LUCIR [7]	63.17	60.14
LUCIR w/ CBAM	65.04	64.53
LUCIR w/ CBAM w/ Info	**67.93**	**66.27**

VI. Remarks

In sum, as we revealed in this paper, our proposed learning scheme could be used to enhance the performance under MCIL settings with various baselines. Yet, our proposed learning scheme is limited on the performance enhancement when comparing with other previous researches, the dual-memory based continual learning

would be applicable to enhance the feature extraction performance and especially overcoming the catastrophic forgetting on the various MCIL settings.

REFERENCES

[1] Davide Abati, Jakub Tomczak, Tijmen Blankevoort, Simone Calderara, Rita Cucchiara, and Babak Ehteshami Bejnordi. Conditional channel gated networks for task-aware continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3931–3940, 2020.

[2] Jihwan Bang, Heesu Kim, YoungJoon Yoo, Jung-Woo Ha, and Jonghyun Choi. Rainbow memory: Continual learning with a memory of diverse samples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8218–8227, 2021.

[3] Sungmin Cha, Hsiang Hsu, Taebaek Hwang, Flavio P Calmon, and Taesup Moon. Cpr: Classifier-projection regularization for continual learning. *arXiv preprint arXiv:2006.07326*, 2020.

[4] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 532–547, 2018.

[5] Sayna Ebrahimi, Franziska Meier, Roberto Calandra, Trevor Darrell, and Marcus Rohrbach. Adversarial continual learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 386–402. Springer, 2020.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[7] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 831–839, 2019.

[8] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[9] Xingjian Li, Haoyi Xiong, Hanchao Wang, Yuxuan Rao, Liping Liu, Zeyu Chen, and Jun Huan. Delta: Deep learning transfer using feature map with attention for convolutional networks. *arXiv preprint arXiv:1901.09229*, 2019.

[10] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

[11] Yaoyao Liu, Bernt Schiele, and Qianru Sun. Adaptive aggregation networks for class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2544–2553, 2021.

[12] Yaoyao Liu, Yuting Su, An-An Liu, Bernt Schiele, and Qianru Sun. Mnemonics training: Multi-class incremental learning without forgetting. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 12245–12254, 2020.

[13] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.

[14] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019.

[15] Ameya Prabhu, Philip HS Torr, and Puneet K Dokania. Gdumb: A simple approach that questions our progress in continual learning. In *European conference on computer vision*, pages 524–540. Springer, 2020.

[16] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.

[17] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[18] Christian Simon, Piotr Koniusz, and Mehrtash Harandi. On learning the geodesic path for incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1591–1600, 2021.

[19] Michalis K Titsias, Jonathan Schwarz, Alexander G de G Matthews, Razvan Pascanu, and Yee Whye Teh. Functional regularisation for continual learning with gaussian processes. *arXiv preprint arXiv:1901.11356*, 2019.

[20] Charles F Van Loan. Generalizing the singular value decomposition. *SIAM Journal on numerical Analysis*, 13(1):76–83, 1976.

[21] Seralynne D Vann and Mathieu M Albasser. Hippocampus and neocortex: recognition and spatial memory. *Current opinion in neurobiology*, 21(3):440–445, 2011.

[22] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

[23] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 374–382, 2019.

[24] Jaehong Yoon, Saehoon Kim, Eunho Yang, and Sung Ju Hwang. Scalable and order-robust continual learning with additive parameter decomposition. *arXiv preprint arXiv:1902.09432*, 2019.

[25] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6023–6032, 2019.

A SUPPLEMENTARY MATERIAL

A. *Generalization performance*

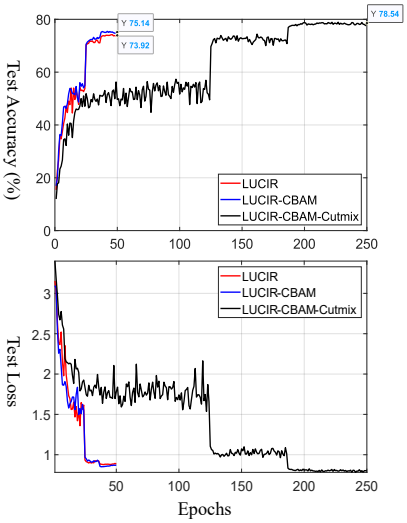


Fig. 5. Loss and accuracy plots over the zeroth training process. As we can see, the cutmix [25] algorithm showed better generalization performance.

As we can see the results described in Figure 5, firstly, attaching the CBAM module slightly enhanced the feature extraction performance at the zeroth learning phase. In the following, by considering the mix-label effects to train the network, we prolonged the set epochs as 250, and as a result, the trained model reached the test accuracy of 78.54(%). Thus, by simply adopting the CutMix algorithm, the proposed method minimized the overfitting effects and achieved more generalized performance. Further, due to this significance, the applied CutMix algorithm showed superior performance on the MCIL phases.

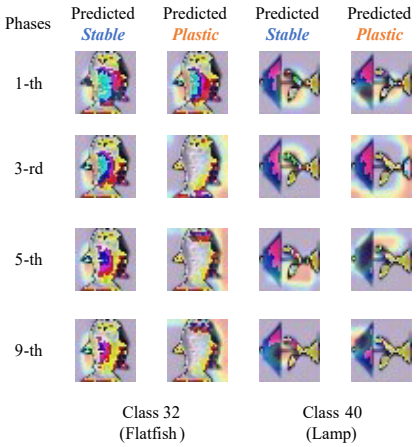


Fig. 6. Heat map implemented randomly selected images from the specific batch by applying guided grad-CAM, by according to the increment of the CL phases.

B. *Grad-CAM results*

To verify the effects of the stable and the plastic blocks, two representative images are shown with grad-CAM [17] with the various phases. As we can see in Figure 6, the results of the visualized results of the input image and its corresponding results at each phase are visualized. The ground-truth label is denoted at the bottom of the figures. As a result, stable blocks could be focused on the informative regions. However, plastic blocks learned on the current task became obsolete to extract meaningful features from the input on the previous tasks.

C. *Variations of fusion variables*

As we can see in Figure 7, the values of fusion variables at each level under multi-phases are represented with different values from each other. Refer to Figure 7, in 10-phase MCIL settings, values of two fusion variables of the plastic and the stable block were quite different on $LUCIR_{CBAM}$ (on the top of figure) and $LUCIR_{CBAM+infoflow}$ (on the bottom of figure). As a result of $LUCIR_{CBAM}$, at the first level, the fusion variable is not converged into the specific value. Conversely, at the second and the third label, the fusion variable diverged and converged into the non-specified space/specific stable point. Conversely, for the case of $LUCIR_{CBAM+infoflow}$, as the searching space of fusion variables is too wide, values of all fusion variables have significantly diverged during the continual learning process. Thus, on level 1, after learning is terminated, the fusion variable of the stable block reached a value bigger than 10.

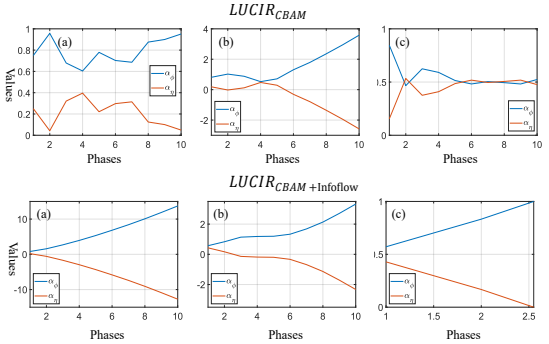


Fig. 7. Variations of fusion variables over the entire 10-phase MCIL settings. In the top, the result of LUCIR with CBAM is shown, and in the bottom, the result of LUCIR with CBAM+Infoflow is represented.

Molecular Graph-based Conditional Variable Autoencoder for De Novo Drug Design

Myeonghun Lee[1] and Kyoungmin Min[2]

[1] School of Systems Biomedical Science, Soongsil University, Seoul, Republic of Korea, leemh216@gmail.com
[2] School of Mechanical Engineering, Soongsil University, Seoul, Republic of Korea, kmin.min@ssu.ac.kr

Abstract

The ultimate goal of drug discovery is to generate molecules with desired properties directly. This study demonstrated a molecular graph-based conditional variable autoencoder (MGC-VAE), which shows that it is possible to generate desired molecules under specific conditions. To do this, drug-like molecules from the ZINC database and partition coefficient (logP) were used to train the model. The performance was compared with graph-based variable autoencoders without conditions.

Keywords— *Conditional variable autoencoder, Drug discovery, Deep learning, Molecular design*

I. INTRODUCTION

Finding new molecules with desired properties for de novo drug design is a significant and challenging task [25]. However, it is almost impossible to find these molecules in the high-dimensional chemical space of all molecules without prior knowledge. Recently, biologists have extensively integrated machine learning algorithms into drug design and discovery processes. Computerized modeling based on machine learning provides a great way to identify and validate compounds, drug toxicity, and drug monitoring [10].

The computer-based molecular design has received much attention as a solution to overcome experimental limitations [28, 26, 6, 24]. High throughput virtual screening is able to find molecules with desired physical properties through fast calculations with high accuracy and low cost. Implementing such a method can significantly reduce time and effort by selecting computationally appropriate molecules from the millions of molecules in the database and then experimentally verifying them [17]. Furthermore, the generative model, which recently emerged with the development of deep learning, encouraged application to new molecular designs that use generative models to propose new molecules likely to have desired properties. Specifically, some successful long short-term memory (LSTM) network architectures have been demonstrated and validated [27, 4, 32, 9].

For a molecular generation, recurrent neural networks (RNN) still play an important role. They were successfully applied to generate SMILES (simplified molecular-input line-entry system) strings, a commonly used text representation of molecules [2, 23]. In particular, RNN architectures based on LSTM achieve excellent results in natural language processing tasks, in which input is a token sequence of different lengths. However, it is unfortunate that creating SMILES expressions of molecules often becomes unstable, resulting in a misconstruction. On the other hand, graphs are more natural data structures for describing molecules, and the advantages of graph generation are being studied in that all outputs are valid molecules [19, 18].

In this study, molecules with the desired log octanol-water partition coefficient (logP) were created using a molecular graph-based conditional variational autoencoder (MGC-VAE). To verify performance, we compared the results of a molecular graph-based variational autoencoder (MGVAE) with the results of using given specific conditions. Furthermore, we determined the probability of valid molecular production depending on the size of the graph and whether it is already included in PubChem [13].

II. RELATED WORKS

One of the most representative specifications used in a molecular generation is SMILES [21]. SMILES is a line notation for describing molecules using strings; thus, text generation models were developed [27, 9, 3]. Some implemented LSTM-based CVAE [17] or reinforcement learning strategies to generate molecules with specific properties [22, 8]. However, text generation models have fundamental limitations in delivering consistent molecular similarities [12]. In contrast, molecular graphs can more naturally express the structural properties of molecules [16]. Most graph generation models use sequential methods. For example, some models produce final graphs, constructing additional nodes and edges [15, 29, 14]. JT-VAE [12] generates a tree of molecular fragments and then determines

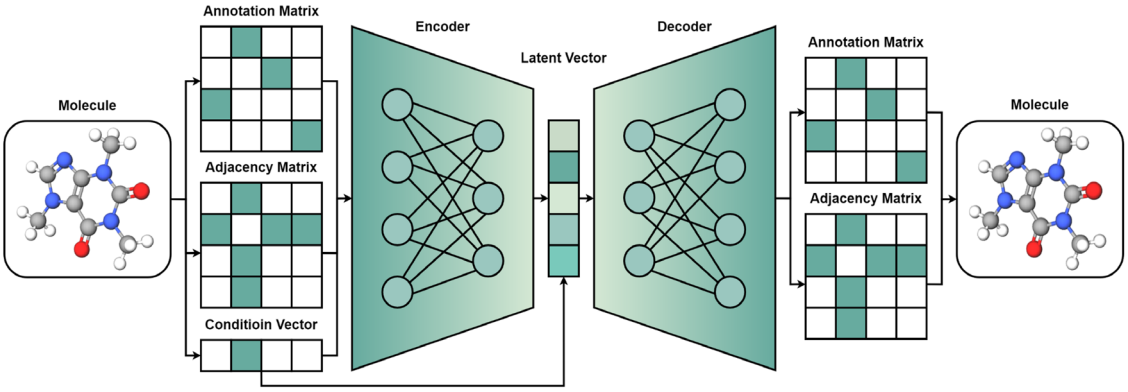


Fig. 1. This is a picture of MGCVAE. Starting with the encoder receiving molecules represented by an annotation matrix and an adjacency matrix with a specific condition vector, the latent space receives a condition vector again and finally the decoder produces a graph of the molecules with specific properties. In addition, VAE does not include the condition vector.

the final molecular graph through the subsequent assembly of the fragments. In addition, GraphRNN [33] is proposed to separate graph generation from hierarchical recurrent neural networks into node and edge sequence generation. In contrast, there is also a way to generate an entire graph. GraphVAE [30] and MolGAN [5] models produce graphs simultaneously; neighboring matrices and graph properties.

III. METHOD

A. *Molecular graphs*

The graph representation of molecules corresponds to atoms by nodes and bonds by edges, represented by an annotation matrix and an adjustment matrix, as shown in Figure 1. The annotation matrix ($N \times N$, N is the number of atoms, X is the number of types of atoms) represents each row as one-hot encoding of atoms, and the adjacency matrix ($N \times N$) represents how each row and column corresponding to the atoms are binding.

B. *Conditional variational autoencoder (CVAE)*

In this work, VAE and CVAE were implemented to generate molecules and compare their performance. VAE produces molecules similar to a given dataset, while CVAE is beneficial in producing molecules similar to a given dataset with a given specific condition, as shown in Figure 1. This is the major difference between the two models due to different objective functions. The objective functions of each VAE (1) and CVAE (2) are as follows:

$$E[\log P(X|z)] - D_{KL}[Q(z|X)||P(z)] \quad (1)$$

$$E[\log P(X|z,c)] - D_{KL}[Q(z|X,c)||P(z|c)] \quad (2)$$

where E represents the expectation value, P and Q indicate the probability distribution, D_{KL} represents the Kullback-Leibler divergence, X, z, and c are the data, latent space, and condition vectors, respectively. $Q(z|X)$ and $P(X|z)$ represent encoder and decoder in autoencoder, respectively.

The main differences between these two models are originated from different objective functions according to condition vector c. In this study, the molecular properties that we want to control have corresponded as condition vectors. As a result, CVAE can generate molecules with target properties imposed by condition vectors. The generative adversarial network (GAN) is not considered in this process because the gradient descent optimization method for the already known properties given to the data is more convenient than the optimization method that offers the desired molecular properties as a reward network by MolGAN [5].

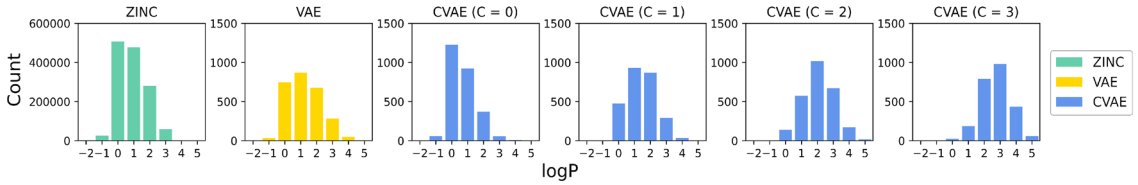
IV. EXPERIMENTS

A. *Dataset*

This study selected 1,354,760 molecules with 14 atoms obtained from the ZINC database [11], as shown in Table 1. Molecules are composed of 12 atoms: B, C, N, O, F, Si, P, S, Cl, Br, Sn, and I. In addition, these molecules also have logP calculated by RDKit [1] greater than –6 and less

Dataset	Molecule	Size	Atom	Bond	Condition
ZINC	1.35M	14	12	4	$-6 < C < 5$

Table 1. The dataset used to train the models. The histogram of the distribution of the data is shown in Figure 3. The Atom and Bond columns indicate the number of each type, and C stands for the condition logP.



Model	Valid Molecules	Condition = 0	Condition = 1	Condition = 2	Condition = 3	In PubChem	Not In ZINC
VAE	53.40 % (2670/5000, 36)	28.05 % (749/2670)	32.66 % (872/2670)	25.36 % (677/2670)	10.56 % (282/2670)	35.62 (951/2670)	91.99 (2456/2670)
CVAE	51.65 % (10330/20000, 659)	46.44 % (1227/2642)	35.70 % (934/2616)	39.35 % (1020/2592)	39.56 % (981/2480)	34.24 (3537/10330)	91.73 (9476/10330)

Fig. 2. Experimental results of VAE and CVAE. It shows the number of molecules that actually have conditions (C, logP) that must be given to generate the molecules. Each models attempted to generate molecules with 5000 samplings from late space, thus CVAEs were tried a total of 20000 times. Among them, the valid molecules contained 36 and 659 duplicates, respectively. These molecules were searched using PubChemPy [31] or converted into InChIKey, a 27-character compacted (hashed) version of InChI (International Chemical Identifier) [20], using RDKit to determine whether they were duplicated or ZINC dataset. (In the case of ZINC, some parts with little data were omitted for visualization.)

than 5, when P is defined as the ratio of the concentrations between two solvents of a solute [7]. This ZINC dataset was divided into a training set and a test set as a ratio of 9:1 to confirm the training process.

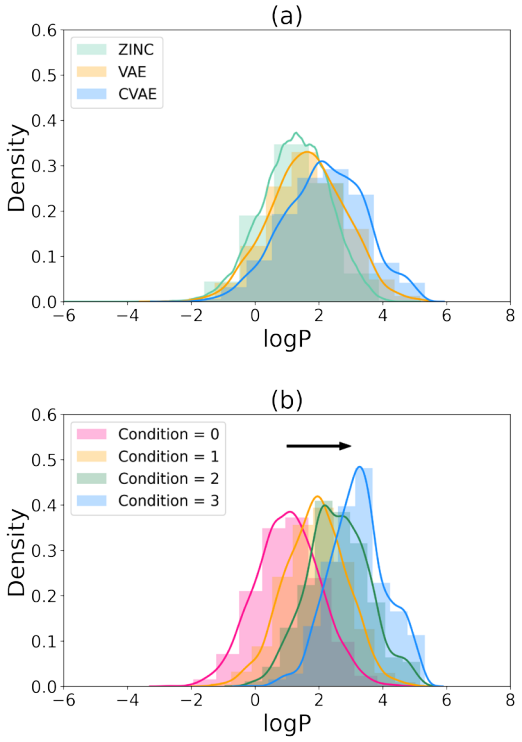


Fig. 3. (a) A histogram of logP for molecules of VAE, CVAE and ZINC dataset. CVAE is located relatively to the right because it contains the case where logP = 3. (b) The histogram according to each conditions for CVAE in (a). As the given condition increases, the histogram gradually moves to the right

B. *Model performance*

The results for the performances of the models can be found in Figure 2. By training logP for molecules of selected ZINC dataset using VAE and CVAE, we determined which logP was generated from VAE that learned the distribution of dataset and CVAE under a specific condition. The two models were trained with the same dataset and generated molecules with 5,000 samples to compare the results. VAE generated 5,000 cases, and CVAE generated molecules with the condition = 0, 1, 2, 3, and 4 cases, making a total of 20,000 attempts. The logP value was calculated only if the generated molecule was valid. In addition, we found that the generated molecules were present in PubChem, and those novel molecules were generated that were not present in the selected ZINC dataset used in training.

The ZINC dataset and VAE had the largest number of molecules with logP = 0, 1, and the smallest number with logP = 3. However, CVAE has the most generated molecules with corresponding values under a given condition, especially logP = 3, which is noticeably more generation than VAE. This means that CVAE can generate molecules for a particular condition. In addition, as shown in Figure 3, (the mean and the standard deviation of logP) for ZINC dataset and the generated molecules were ZINC (1.25, 1.07), VAE (1.63, 1.19), and CVAE (2.16, 1.26), respectively. CVAE has a relatively large value, including the case of condition = 3, because the ZINC dataset has fewer logP = 3. Furthermore, as a result of each condition of CVAE, the mean of logP for the generated molecules also increased to 1.02, 1.89, 2.58, and 3.22, when condition = 0, 1, 2, 3, respectively.

The number of atoms is associated with the size of the annotation matrix and the adjacency matrix. The larger the size of the molecules, the greater the size of the matrixes the model should generate, increasing the probability of

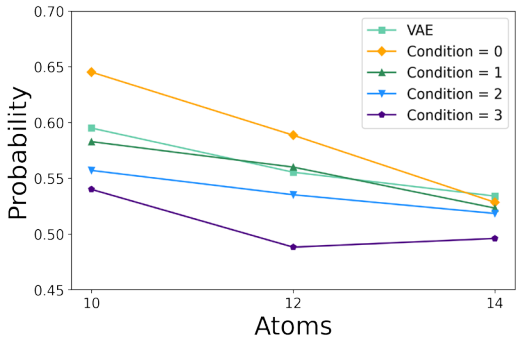


Fig. 4. Probability of vaild molecular generation based on number of atoms.

generating invalid graphs. Thus, 14 atoms often produce invalid outputs where generated matrixes are more difficult to convert as molecules than 10 atoms case. For this reason, as shown in Figure 4, reducing the size of the molecules to a maximum of 10 showed a higher success rate in all cases than in 14 atoms case. On the other hand, in the case of 10 atoms, the amount of ZINC dataset used in training decreased to 83,017, with 12 using 397,581 data. As a result, for 10 atoms case, fewer molecules were used in training, but more were actually converted.

V. CONCLUSION

In this study, we created molecules using Graph-based VAE and CVAE, and we demonstrated whether molecules with desired specific properties are generated from CVAE. It indicates that the two models were able to generate molecules, and CVAE produced the most desired molecules. However, Graph-based CVAEs are more difficult to produce than LSTM-based CVAE due to the limited number of atoms to be implemented. In addition, multivariable control is expected to be possible compared to LSTM-based CVAE [17].

REFERENCES

[1] The RDKit Documentation — The RDKit 2021.03.1 documentation.

[2] Josep Arús-Pous, Thomas Blaschke, Silas Ulander, Jean-Louis Reymond, Hongming Chen, and Ola Engkvist. Exploring the GDB-13 chemical space using deep generative models. *Journal of Cheminformatics*, 11(1), March 2019.

[3] Esben Bjerrum and Boris Sattarov. Improving chemical autoencoder latent space and molecular de novo generation diversity with heteroencoders. *Biomolecules*, 8(4):131, October 2018.

[4] Esben Jannik Bjerrum and Richard Threlfall. Molecular generation with recurrent neural networks (rnns), 2017.

[5] Nicola De Cao and Thomas Kipf. Molgan: An implicit generative model for small molecular graphs, 2018.

[6] Tiejun Cheng, Qingliang Li, Zhigang Zhou, Yanli Wang, and Stephen H. Bryant. Structure-based virtual screening for drug discovery: a problem-centric review. *The AAPS Journal*, 14(1):133–141, January 2012.

[7] John Comer and Kin Tam. Lipophilicity profiles: Theory and measurement. In *Pharmacokinetic Optimization in Drug Research*, pages 275–304. Verlag Helvetica Chimica Acta.

[8] Gabriel Lima Guimaraes, Benjamin Sanchez-Lengeling, Carlos Outeiral, Pedro Luis Cunha Farias, and Alán Aspuru-Guzik. Objective-reinforced generative adversarial networks (organ) for sequence generation models, 2017.

[9] Anvita Gupta, Alex T. Müller, Berend J. H. Huisman, Jens A. Fuchs, Petra Schneider, and Gisbert Schneider. Generative recurrent networks for de novo drug design. *Molecular Informatics*, 37(1-2):1700111, November 2017.

[10] Rohan Gupta, Devesh Srivastava, Mehar Sahu, Swati Tiwari, Rashmi K. Ambasta, and Pravir Kumar. Artificial intelligence to deep learning: machine intelligence approach for drug discovery. *Molecular Diversity*, 25(3):1315–1360, April 2021.

[11] John J. Irwin and Brian K. Shoichet. Zinc - a free database of commercially available compounds for virtual screening. *Journal of Chemical Information and Modeling*, 45(1):177–182, 2005. PMID: 15667143.

[12] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation, 2018.

[13] Sunghwan Kim, Paul A. Thiessen, Evan E. Bolton, Jie Chen, Gang Fu, Asta Gindulyte, Lianyi Han, Jane He, Siqian He, Benjamin A. Shoemaker, Jiyao Wang, Bo Yu, Jian Zhang, and Stephen H. Bryant. PubChem substance and compound databases. *Nucleic Acids Research*, 44(D1):D1202–D1213, September 2015.

[14] Yibo Li, Liangren Zhang, and Zhenming Liu. Multi-objective de novo drug design with conditional graph generative model. *Journal of Cheminformatics*, 10(1), July 2018.

[15] Yujia Li, Oriol Vinyals, Chris Dyer, Razvan Pascanu, and Peter Battaglia. Learning deep generative models of graphs, 2018.

[16] Jaechang Lim, Sang-Yeon Hwang, Seokhyun Moon, Seungsu Kim, and Woo Youn Kim. Scaffold-based molecular design with a graph generative model. *Chemical Science*, 11(4):1153–1164, 2020.

[17] Jaechang Lim, Seongok Ryu, Jin Woo Kim, and Woo Youn Kim. Molecular generative model based on conditional variational autoencoder for de novo molecular design. *Journal of Cheminformatics*, 10(1), July 2018.

[18] Łukasz Maziarka, Agnieszka Pocha, Jan Kaczmarczyk, Krzysztof Rataj, Tomasz Danel, and Michał Warchoł. MolCycleGAN: a generative model for molecular optimization. *Journal of Cheminformatics*, 12(1), January 2020.

[19] Rocío Mercado, Tobias Rastemo, Edvard Lindelöf, Günter Klambauer, Ola Engkvist, Hongming Chen, and Esben Jannik Bjerrum. Graph networks for molecular design. *Machine Learning: Science and Technology*, 2(2):025023, March 2021.

[20] Igor Pletnev, Andrey Erin, Alan McNaught, Kirill Blinov, Dmitrii Tchekhovskoi, and Steve Heller. InChIKey collision resistance: an experimental testing. *Journal of Cheminformatics*, 4(1), December 2012.

[21] Daniil Polykovskiy, Alexander Zhebrak, Benjamin Sanchez-Lengeling, Sergey Golovanov, Oktai Tatanov, Stanislav Belyaev, Rauf Kurbanov, Aleksey Artamonov, Vladimir Aladinskiy, Mark Veselov, Artur Kadurin, Simon Johansson, Hongming Chen, Sergey Nikolenko, Alán Aspuru-Guzik, and Alex Zhavoronkov. Molecular sets (MOSES): A benchmarking platform for molecular generation models. *Frontiers in Pharmacology*, 11, December 2020.

[22] Mariya Popova, Olexandr Isayev, and Alexander Tropsha. Deep reinforcement learning for de-novo drug design. 2017.

[23] Mariya Popova, Olexandr Isayev, and Alexander Tropsha. Deep reinforcement learning for de novo drug design. *Science Advances*, 4(7):eaap7885, July 2018.

[24] Jean-Louis Reymond, Ruud van Deursen, Lorenz C. Blum, and Lars Ruddigkeit. Chemical space as a source for new drugs. *MedChemComm*, 1(1):30, 2010.

[25] Gisbert Schneider and Uli Fechner. Computer-based de novo design of drug-like molecules. *Nature Reviews Drug Discovery*, 4(8):649–663, August 2005.

[26] Thomas Scior, Andreas Bender, Gary Tresadern, José L. Medina-Franco, Karina Martínez-Mayorga, Thierry Langer, Karina Cuanalo-Contreras, and Dimitris K. Agrafiotis. Recognizing pitfalls in virtual screening: A critical review. *Journal of Chemical Information and Modeling*, 52(4):867–881, 2012. PMID: 22435959.

[27] Marwin H. S. Segler, Thierry Kogej, Christian Tyrchan, and Mark P. Waller. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS Central Science*, 4(1):120–131, December 2017.

[28] Brian K. Shoichet. Virtual screening of chemical libraries. *Nature*, 432(7019):862–865, December 2004.

[29] Gregor N. C. Simm and José Miguel Hernández-Lobato. A generative model for molecular distance geometry. 2019.

[30] Martin Simonovsky and Nikos Komodakis. Graphvae: Towards generation of small graphs using variational autoencoders, 2018.

[31] Matt Swain. Pubchempy documentation¶.

[32] Robin Winter, Floriane Montanari, Frank Noé, and Djork-Arné Clevert. Learning continuous and data-driven molecular descriptors by translating equivalent chemical representations. *Chemical Science*, 10(6):1692–1701, 2019.

[33] Jiaxuan You, Rex Ying, Xiang Ren, William L. Hamilton, and Jure Leskovec. Graphrnn: Generating realistic graphs with deep auto-regressive models, 2018.

Single CNN-based Fall Detection by Cut and Paste Data Augmentation

Sunhee Hwang[1], Minsong Ki[1], Seung-Hyun Lee[1] and Sanghoon Park[1]

[1] LG Uplus Corp., sunheehwang@lguplus.co.kr

Abstract

Deep learning based fall detection is one of the crucial tasks for intelligent video surveillance systems, which aims to detect unintentional falls of humans and alarm dangerous situations. On the other hand, due to the limited environment and the small number of abnormal samples, the existing methods achieve their goals by using external knowledge or large-size models. In this work, unlike priors, we introduce to classify falls through a single and small size convolutional neural network. To this end, we introduce a new data augmentation method to generate a large-scale synthetic fall detection dataset for model training. At the inference step, we simply apply pre-processing method for input frames, then classify it through the small sized model. In experiment, with the qualitative and quantitative evaluations on URFD and AIHub airport datasets compared to the existing methods, we demonstrate the effectiveness of our method.

Keywords— *Real-time Fall Detection, Video Surveillance, Image Blending, Deep Neural Networks*

I. INTRODUCTION

Falling is one of the risky factors causing unintentional injury, particularly for the elderly, lone workers, and patients. According to the WHO [23], approximately 684,000 fatal falls occur, resulting in deaths each year globally. To enable quick responses to such accidents, automated monitoring systems are getting widely used in nursing homes, manufacturing industries, and hospitals. Specifically, a variety of fall detection methods are proposed to detect the human falling status based on diverse kinds of sensors, *e.g.*, accelerometers, gyroscopes, biometric, and visual sensors. Among these, vision-based method is an effective way to apply it to existing video surveillance systems in our daily lives, since the systems are non-invasive and there is no hassle of wearing devices.

In recent years, deep learning based fall detection systems for video surveillance have achieved great performance. Most of them adopt a multiple-stage-based approach for detecting falls by training spatiotemporal information, which consist of two or more modules [7, 14, 4, 5, 11]. They firstly detect person or its region using pre-trained CNNs [8, 3, 20], then classify its sequential information is falling state or not. These methods have an advantage of producing excellent performance by utilizing external knowledge from additional modules, which are trained on large-scale datasets. While, since a certain module from multiple-stage-based approach can affect the overall performance and inference time. This makes a difficulty of maintaining the surveillance system, since, in case of the failure to detect falls, each module have to be analyzed and updated.

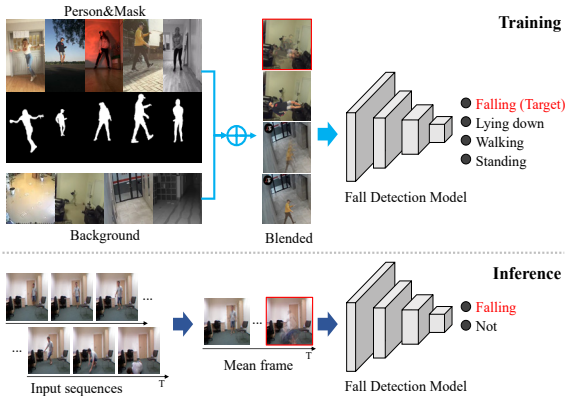


Fig. 1. Overview of the proposed method.

Several approaches simplify the fall detection framework by based on single convolutional neural networks (CNNs) [2, 15, 18, 17]. Given input video, they utilize the raw images or apply simple pre-processing to represent human motion from sequences. Then, they predict the falling state using single CNNs rather than using additional modules. These are simple and easy way to apply the fall detection method to the real-world intelligent video surveillance systems. On the other hand, most available fall detection datasets are small-scale, which includes the limited number of person or place. With the small training dataset, it is not easy to obtaining outstanding performance without utilizing external knowledge. As an alternative way, several methods [2, 18] use large-scale CNNs such as VGG16.

From the existing works, we observe that the fall de-

(a) Person & Mask (b) Background (c) r_1 (d) r_2 (e) r_3 (f) r_N (g) $avg(\mathbf{r})$

Fig. 2. Procedure of our image blending method. Given person&mask (a) and background image (b), we cutout person region from (a) and paste it into (b) after applying transform **t**: (c)-(f). Then, we average the images (c)-(f): (g). Finally, (f) and (g) are utilized to train the fall detection model as motionless and motional images, respectively.

tection models trained with large amounts of data or large sized models tend to outperform than others. In this work, we aim to achieve great performance with the small sized model by proposing a new method to address the problem of dataset insufficiency. To this end, inspired by [13, 6], we introduce a data augmentation strategy to generate diverse samples for model training. The overall process of our method is shown in Figure 1. In the training step, given person and background images, our image synthesis method generate a great number of falling/non-falling samples, which does not require acting real dangerous behaviors for obtaining fall datasets. Then, we train the fall detection model with the generated images. At the inference step, instead of extracting pose and optical flow to represent human motion, we simply calculate the average frame of a certain frames to create a single human motion image. Accordingly, our method enables real-time fall detection using a single CNN based model.

In experiment, we verify the superiority of our method with previous state-of-the-art methods in terms of accuracy and procedures on URFD dataset [12]. We also compare the inference speed, and accuracy among various single CNNs. Overall, the proposed approach achieves the best performance on EfficientNet-B0 [22] with the fall detection accuracy of 97.14% and 92.25% on URFD and AIHub airport dataset [1], respectively.

In summary, our main contributions are:

- We propose a simple yet effective fall detection framework through the proposed image blending and augmentation methods, reducing the hardware and dataset requirements.
- Through the quantitative and qualitative results on URFD and AIHub datasets, we validate the effectiveness of the proposed method.

[1]This study was conducted using the airport abnormal behavior CCTV dataset constructed as part of the 「2019 Artificial Intelligence Identification and Tracking System Construction」 project of National IT Industry Promotion Agency (NIPA). Utilized data can be obtained from AI Hub (https://aihub.or.kr/node/6258)

II. Proposed Method

A. *Image Synthesis*

We describe our novel image blending method that generates a large number of falls images to solve the problem of insufficient training data. Specifically, we produce two types of human motion images as falling and walking. Also, we make two motionless images as lying down and standing. The image blending is composed of two parts: generating sequential images of human motion and averaging the generated sequential images. Figure 2 shows the overall procedure of our image synthesis.

Let a person image as x and the corresponding binary mask as m. Before generating synthetic human motion images, we apply an online data augmentation approach with image transformation f to x, where f is composed of different kinds of randomly selected image jittering functions to ensure the diversity of the training images. Afterwards, given $\hat{x} = f(x)$, we cutout the person segments using m, and then apply geometrical transformation $\mathbf{t} \in \{t_1, t_2, ..., t_N\}$, where $\mathbf{t}$ and N denote a set of transformation functions to generate continuous human motion and the number of moments for the motion, respectively. In other words, the transformation $\mathbf{t}$ is designed to reflect the falling or walking motion, resulting in the N sequential images. Then, we individually paste the N images, *i.e.*, $t_1(\hat{x}), t_2(\hat{x}), ..., t_N(\hat{x})$, into background image b. As a results, we attain the blended images $\mathbf{r} \in \{r_1, r_2, ..., r_N\}$ representing a series of moment for human motion. While our goal is to train a single CNN without utilizing additional sequential modules, thus, we average $\mathbf{r}$ (a set of N blended images) to represent human motion in a single image. Finally, we utilize $r_N \in \{lying\ down, standing\}$ and $avg(\mathbf{r}) \in \{falling, walking\}$ for training samples of motionless image and motional image, respectively.

Table 1. Fall detection performance comparison between our approach and exiting methods on the URFD dataset. The Second column denotes the architecture is composed of a single CNN or not. The third column indicates the usage of URFD dataset for model training, where D_{tr} denotes the training dataset. We also note the number of parameters of single CNN-based models.

Ref.	Single CNN	URFD in D_{tr}	Architecture	Sensitivity	Specificity	Precision	Accuracy	Params
[7]			P-CNN + Tensorized-LSTM	**100**	97.44	-	**99.00**	-
[14]			OpenPose + LSTM	**100**	96.40	-	98.20	-
[4]			Mask-RCNN + VGG16 + Bi-LSTM	91.80	**100**	**100**	96.70	-
[5]			YOLO V3 + DeepSort + LSTM	93.10	-	94.80	-	-
[11]			PifPaf + Rule-based	83.33	**100**	-	-	-
[2]			VGG16	-	-	-	92.34	138M
[15]			BGS + LeNet	88.00	-	89.00	89.00	60K
[18]			Modified VGG16	**100**	92.00	-	95.00	138M+
[17]			Shallow-CNN	41.47	-	31.04	88.55	104K
Ours			EfficientNet-B0	93.33	**100**	**100**	**97.14**	4M

B. *Falling State Classification*

To predict falls, our model is trained to classify four defined human behavior (*i.e.*, lying down, standing, falling, and walking), which appears in the actual monitoring environment. To this end, we employ the online data augmentation with our image blending method to compose a mini-batch for training. Let $s^i \in \{r_N, avg(\mathbf{r})\}$ is one of the randomly generated human behavior image. We compose the mini-batch with M images $\mathbf{s} \in \{s^1, s^2, ..., s^M\}$. Then, we train a single CNN based fall detection model g to predict corresponding human behavior label $\mathbf{y} \in \{y^1, y^2, ..., y^M\}$, where g is optimized by the following cross-entropy loss function:

$$\mathscr{L}_{ce} = -\sum_{i=1}^{M} y^i log(g(s^i)). \quad (1)$$

For inference, given real images from a camera, *i.e.*, $\mathbf{u} \in \{u^1, u^2, ..., u^\infty\}$, we start detecting falls from time T using N selected frames with a stride k. For example, given $\hat{\mathbf{u}} \in \{u^{T-k\times N}, u^{T-k\times(N-1)}, ..., u^T\}$, we calculate the average of it. Afterwards, we predict the target of human status using g from $avg(\hat{\mathbf{u}})$.

III. EXPERIMENTS

A. *Experimental Settings*

*1) Image blending:*For the image blending, we utilize two datasets of HDD (human detection dataset) [2], and TikTok dancing dataset [3] from Kaggle. HDD dataset consists of 236 different scenes without humans. For this, we select ten background images in which no object is placed in the center of it. In the TikTok dataset, there are 2,615 person images with corresponding binary masks, where all images are utilized to generate a training dataset. By blending images from both datasets with random image transformation, we obtain a tremendous amount of images for training.

[2]https://www.kaggle.com/constantinwerner/human-detection-dataset

[3]https://www.kaggle.com/tapakah68/segmentation-full-body-tiktok-dancing-dataset

Table 2. Performance of prediction intervals on URFD.

Interval		Accuracy
Second	Frames	
0.4	10	90.0
0.6	15	95.71
0.8	20	95.71
1.0	25	**97.14**
1.2	30	91.43
1.4	35	91.43
1.6	40	85.71

*2) Training and evaluation:*To train the model, we only use the synthetic images generated by our image blending method. For the evaluation, we use two different benchmark datasets: URFD [12] and AIHub abnormal behavior in airport CCTV. URFD dataset consists of 30 falls and 40 adl (activity of daily living) clips. The frame rate is 25 fps and it has a short duration, the average duration of fall videos is under 5 seconds. All videos of URFD are utilized to evaluate our method at video-level. To this end, we regard a video including a motion image predicted to be falling at least once as falling, and consider a video never classified as a falling state as non-falling. Experimentally, the interval for the detection set to 1 second. AIHub abnormal behavior in airport CCTV dataset consists of 10k clips with three different recording environment. Among those, we utilize fall and normal videos from *cubox* anomaly scene, which consist of 348 fall clips and 2,109 normal clips with longer duration than URFD *i.e.*, 10 seconds or longer with a higher frame rate as 30. For this data, the interval for the detection set to 5 seconds, empirically.

*3) Implementation:*The two datasets have different environment settings of camera position, the distance between camera and person, and the duration of clips. Thus, we train the models with two different image blending settings, *i.e.*, scale and transition of person image. For the image blending, we utilize the OpenCV and albumentation [1]. Fall detection models are trained about 100 epochs using an AdamW [16] optimizer with a learning rate of 0.001 on a single GPU (RTX 3080), which are implemented using PyTorch library [19].

(a) Falling (b) Non-falling

Fig. 3. Examples of mean frame. Row 1-2 and 3 denote examples of URFD and AIHub dataset, respectively. Examples of last column are the failure cases.

Table 3. The computational speed of CNNs on URFD.

Model	Frames per second		Accuracy
	Pytorch	ONNX	
ResNet-18 [9]	**707.01**	**1092.80**	94.26
MnasNet [21]	472.22	1017.88	92.86
MobileNetV3 [10]	320.34	722.93	94.29
EfficientNet-B0 [22]	162.46	204.04	**97.14**

Table 4. Fall Detection results on AIHub dataset.

	Frame-level Accuracy
Sensitivity	71.15
Specificity	95.76
Precision	73.59
Accuracy	92.25

B. *Results*

*1) Evaluation on URFD dataset:*Table 1 shows the fall detection performance of our model compared to existing methods [7, 14, 4, 5, 11, 2, 15, 18, 17]. The top five rows are the results of multiple-stage-based approaches, and the other rows are the results of single CNN-based methods. The proposed method obtains comparable performance to the state-of-the-art methods using multiple modules, in terms of accuracy. Besides, our method achieves the best performance among single CNN-based methods using EfficientNet-B0 [22]. We note that our method achieved best performance on EfficientNet-B0. Table 2 shows the results of ablation study on the prediction intervals for calculating mean frames. Among the intervals, we achieved the best result in one second. We also compare the computational speed and accuracy on following low-cost CNNs: ResNet-18 [9], MnasNet [21], MobileNetV3 [10], EfficientNet-B0, as shown in Table 3. We demonstrate that our method has a slight performance degradation although adopting a model for high speed.

*2) Evaluation on AIHub dataset:*We also evaluate on AIHub dataset, as shown in Table 4. The results are estimated by frame-level since the falling frames are partial ($< 50\%$) of whole video clips. For this dataset, our method achieves an accuracy of 92.25% for frame-level detection.

*3) Qualitative results:*Figure 3 shows the examples of URFD and AIHub datasets predicted as falling or not. As expected, the case similar to the image generated through our image blending method was detected well, but failed in the case of falls, which is difficult to be represented only with rotation or transition transformation.

IV. Conclusion

In this paper, we introduce a novel fall detection method for real-time processing using only a single CNN, which takes a single frame as input. To this end, we propose a simple image blending method to generate human motional and motionless images, which is utilized to build a large size dataset for model training. At the inference step, our method calculates the mean frame of a defined sequence and uses it as input of the detection model without using an additional time-consuming module or operation. Through the extensive evaluation on URFD and AIHub airport datasets, we show the proposed method has the best performance among baselines.

References

[1] Alexander Buslaev, Vladimir I Iglovikov, Eugene Khvedchenya, Alex Parinov, Mikhail Druzhinin, and Alexandr A Kalinin. Albumentations: fast and flexible image augmentations. *Information*, 11(2):125, 2020.

[2] Xi Cai, Xinyue Liu, Suyuan Li, and Guang Han. Fall detection based on colorization coded mhi combining with convolutional neural network. In *2019 IEEE 19th International Conference on Communication Technology (ICCT)*, pages 1694–1698. IEEE, 2019.

[3] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *IEEE transactions on pattern analysis and machine intelligence*, 43(1):172–186, 2019.

[4] Yong Chen, Weitong Li, Lu Wang, Jiajia Hu, and Mingbin Ye. Vision-based fall event detection in complex background using attention guided bi-directional lstm. *IEEE Access*, 8:161337–161348, 2020.

[5] Qi Feng, Chenqiang Gao, Lan Wang, Yue Zhao, Tiecheng Song, and Qiang Li. Spatio-temporal fall event detection in complex scenes using attention guided lstm. *Pattern Recognition Letters*, 130:242–249, 2020.

[6] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D. Cubuk, Quoc V. Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2918–2928, June 2021.

[7] Ziyi Guan, Shuwei Li, Yuan Cheng, Changhai Man, Wei Mao, Ngai Wong, and Hao Yu. A video-based fall detection network by spatio-temporal joint-point model on edge devices. In *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 422–427. IEEE, 2021.

[8] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[10] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1314–1324, 2019.

[11] Bogdan Kwolek and Michal Kepski. Human fall detection on embedded platform using depth maps and wireless accelerometer. *Computer methods and programs in biomedicine*, 117(3):489–501, 2014.

[12] Bogdan Kwolek and Michal Kepski. Human fall detection on embedded platform using depth maps and wireless accelerometer. *Computer methods and programs in biomedicine*, 117(3):489–501, 2014.

[13] Chun-Liang Li, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister. Cutpaste: Self-supervised learning for anomaly detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9664–9674, June 2021.

[14] Chuan-Bi Lin, Ziqian Dong, Wei-Kai Kuan, and Yung-Fa Huang. A framework for fall detection based on openpose skeleton and lstm/gru models. *Applied Sciences*, 11(1):329, 2021.

[15] Wei Liu, Jie Guo, Zheng Huang, and Weidong Qiu. Falling-action analysis algorithm based on convolutional neural network. In *Proceedings of the International Conference on Communication and Electronic Information Engineering (CEIE)*. Atlantis Press, 2016.

[16] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[17] Carlos Menacho and Jhon Ordoñez. Fall detection based on cnn models implemented on a mobile robot. In *2020 17th International Conference on Ubiquitous Robots (UR)*, pages 284–289. IEEE, 2020.

[18] Adrián Núñez-Marcos, Gorka Azkune, and Ignacio Arganda-Carreras. Vision-based fall detection with convolutional neural networks. *Wireless communications and mobile computing*, 2017, 2017.

[19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019.

[20] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[21] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V Le. Mnasnet: Platform-aware neural architecture search for mobile. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2820–2828, 2019.

[22] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019.

[23] Fact Sheets (WHO). Falls. `https://www.who.int/news-room/fact-sheets/detail/falls`, 26 April, 2021.

Survey on Graph Attention Networks for Computer Vision Research

Chanwoo Kim[1], Hoseong Cho[2], Hansol Lee[3], Yungseong Cho[4] and Seungryul Baek[4]

[1] Kyungpook National University, Mathematics, Daegu, South Korea, sky9739a@gmail.com
[2] Inha University, Information and Communication Engineering, Incheon, South Korea, whghtjd9500@gmail.com
[3] UNIST, Computer Science and Engineering, Ulsan, South Korea, hansollee@unist.ac.kr
[4] UNIST, AI Graduate School, Ulsan, South Korea, {ysch0710, srbaek}@unist.ac.kr

Abstract

Graph convolutional network (GCN) was introduced to learn the end-to-end mapping between input and output for the graph-structured data. In the computer vision field, there have been a variety of data types; among them, graph-structured data and the graphical relationship among data have recently been tackled thanks to the introduction of such graph-based approaches. Vanilla GCN use the spectral graph convolution operation which has disadvantages for its computation and the limited application domain that spans only on fixed and un-directed graphs. GCN using the spatial graph convolution provide more general usage and have been widely used in the computer vision applications. Recently, attention mechanism is introduced and the graph attention networks (GATs) are proposed to reinforce more important nodes while under-weight less important nodes, by assigning the importance according to relevance to each neighbor during calculating. Furthermore, the approach generally achieved better performance than Vanilla GCN. In this survey paper, we investigated diverse GAT-based approaches applied in the computer vision field.

Keywords— *Deep Learning, Graph Attention Networks, Computer Vision*

I. Introduction

There have been various computer vision tasks such as image classification, object detection, semantic segmentation, pose estimation and so on. Recently, CNNs have been the state-of-the-art methods for such vision tasks whose input images are in the Euclidean space. The issue arises when the non-Euclidean structured data (e.g. 3D mesh, 3D skeletons or graphical relationship among data) are input to the CNN architecture; while in the computer vision field, making good use of such diverse data format and the graphical relationship among data have known to be important to achieve the good accuracy.

To deal with the graph-structured data, graph neural network (GNN) [12] have been appeared, which is proposed to perform the convolution operations on graph-structured data. There have been proposed two types of the graph convolution operations: the spectral method and the non-spectral method. In [1], the graph-convolution operation is defined in the Fourier domain using the eigen-decomposition on graph Laplacians which is the spectral methods. As obtaining eigenvectors of Laplacian is cumbersome, in [4], a K-order Chebyshev polynomial that approximates smooth filters in the spectral domain was proposed. The most commonly used GNN is the graph convolutional netework (GCN) proposed in [10], which simplified the above methods by defining filters using the first-order approximation and introducing layer-wise propagation.

The issues remained in the spectral approaches are that they rely heavily on the graph Laplacian, which is computationally expensive. Furthermore, as GCN apply the same weight to all neighboring nodes during processing the convolution operation, it lacks the generalization ability: a model learned in one domain cannot be directly applied to another domain. To relieve such issues, graph attention networks (GATs) [14] integrated a non-spectral graph convolution method and an attention mechanism for assigning different weights to neighborhoods. Furthermore, GATs can be parallelized across node-neighbor pairs, so they are computationally more efficient than GCN.

GATs are the most recently introduced graph-based framework and their use in the computer vision field might be of interests to other researchers. Therefore, we conducted a survey on applications using GATs in this paper. In our survey, we include many computer vision applications such as image classification [13], pose estimation [3], recognition [6, 7, 18], few-shot learning [8], visual tracking [5], image generation [11] and some of 3D tasks dealing with point clouds [2, 15]. During the survey, we focused on the role of GATs in each application and the benefits authors obtained from GATs.

II. Base Knowledge & Vanilla GCN

Before explaining GATs, this section briefly describes the terms which are pre-required understanding GCN.

A. Definition

The graph structure is mainly expressed as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{G}$ is a graph, $\mathcal{V}$ is a set of nodes, and $\mathcal{E}$ is a set of edges. This is used to create an Adjacency Matrix A, and A_{ij} describes the connectivity between node i and node j. Therefore, matrix A has a symmetric property in case of an un-directed graph. Also, Laplacian Matrix, which is often used in calculations, is expressed in the form $L = A - D$, where D is a diagonal matrix expressing the degree of each node in the diagonal entry. And $\sigma(\cdot)$ is an activation function, $H^{(l)}$ is the matrix of activation in l^{th} layer; $H^{(0)} = X$.

B. Vanilla GCN

This section introduces layer-wise propagation using spectral-based convolution introduced in [10] among graph-based neural network models. The theoretical motivation of multi-layer graph convolutional network (GCN) will be provided.

This uses the spectral convolution in the graph defined by the product of the filter and the signal, and at this time, using the matrix obtained by applying the eigen-decomposition to the normalized Laplacian, the formula is as follows:

$$g_\theta \star x = U g_\theta U^\top x, \tag{1}$$

where U is the matrix of eigenvectors of the normalized graph Laplacian $L = I_N - D^{-\frac{1}{2}} A D^{-\frac{1}{2}} = U \Lambda U^\top$, with a diagonal matrix of its eigenvalues Λ and $U^\top x$ being the graph Fourier transform of x. Then to reduce time complexity, Chebyshev polynomial is introduced and only K-th neighbor nodes are reflected in the filter approximation. The obtained expression is as follows:

$$g_{\theta'}(\Lambda) \approx \sum_{k=0}^{K} \theta'_k T_k(\tilde{\Lambda}), \tag{2}$$

where x is signal, g is filter, and $\tilde{\Lambda} = \frac{2}{\lambda_{max}} - I_N$. And λ_{max} is largest eigenvalue of normalized graph Laplacian, θ' is Chebyshev coefficients. Here, let λ_{max} = 2, K = 1, and to reduce the number of parameters, set $\theta'_0 = \theta'_1$. And, for numerical stability, the final equation created by applying the renormalization trick is as follows:

$$Z = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} X \Theta. \tag{3}$$

In [10], a 2-layer GCN was created using this and used for semi-supervised node classification, and the formula is as follows:

$$Z = f(X, A) = \mathrm{softmax}\Big(\hat{A}\ \mathrm{ReLU}\Big(\hat{A} X W^{(0)}\Big) W^{(1)}\Big). \tag{4}$$

In here, $\hat{A} = \tilde{D}^{-1/2} \tilde{A} \tilde{D}^{1/2}$ (Renormalization trick) and $W^{(0)}$ is input-to-hidden weight, $W^{(1)}$ is hidden-to-output weight matrix.

III. Graph Attention Networks

Next, we will introduce the attention network. GATs [14] use the non-spectral method that directly defines convolutions only for neighboring nodes. The model uses the attention method for node classification in graph-structured data, which uses the self-attention mechanism to calculate the hidden representation of a node with a weight on each neighbor. First, the attention coefficients are calculated using the shared weight matrix to perform the self-attention on the node. Here, the normalization is performed by softmax and LeakyReLU is used as a nonlinear activation. Also, masked attention is utilized for calculating attention coefficients of neighborhood from center node. The formula for calculating the attention coefficient and the final output are as follows:

$$\alpha_{ij} = \mathrm{softmax}_j(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}_i} \exp(e_{ik})}. \tag{5}$$

$$\vec{h}'_i = \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} \mathbf{W} \vec{h}_j\right). \tag{6}$$

Additionally, the knowledge where the multi-head attention is useful was found, so as to the multi-head attention is proposed. This is a method that concatenates all outputs using the K independent attention mechanism and the formula is:

$$\vec{h}'_i = \Big\Vert_{k=1}^{K} \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^k \mathbf{W}^k \vec{h}_j\right) \tag{7}$$

Finally, the multi-head attention heads in the final layer reflected the non-linearity after averaging, as the concatenation is no longer meaningful. Since the method presented by GATs did not require eigen-decomposition or high cost matrix operations and can assign different importance to each neighbor, it is efficient.

After the introduction of GATs, few more updates have been proposed in the GATs. We will introduce two representative architectures here. The first is gated attention networks (GaAN) [19]. The multi-head attention in GATs have been simply concatenated and did not include the importance of each head. The GaAN explained the content that assigns more weight on the more important heads. For the method explained by GaAN, the linear projection was applied for acquiring query vector from center node feature, and key, value vectors from neighboring node feature. After that, authors used the multiple head attention mechanism. At this time, the soft gate, which is a value between 0 and 1, was calculated and used to give each head another importance.

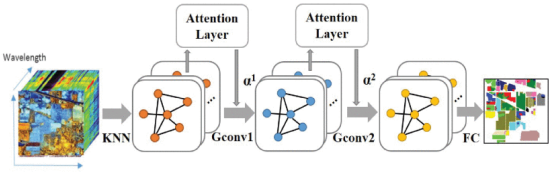


Figure 1. The architecture of the spatial-spectral Graph Attention networks (**S^{2}GATs**)

Second is the sparse graph attention networks (SGAT) [17] which presents a method to give only one attention to each edge in the GATs architecture. A problem with GATs was that since multi-head attention learned very similar attention coefficients, and multi-heads in each layer generated various attention coefficients on the same edge, it was impossible to assign a unique attention score to each edge. SGAT used L_0-norm regularization to remove noisy and task-irrelevant edges to ensure that the GATs model uses the fewest possible edges. The main idea of SGAT was to connect a binary gate to each edge and decide whether to use it for neighbor aggregation. Authors presented the figure of distribution for the analysis of the attention coefficient, which showed that GATs learn a similar distribution of attention scores from all heads and all layers.

IV. Application

In this section, we introduced many applications where GATs have been applied so far.

A. *Image Classification*

GATs [13] have been used for semi-supervised image classification tasks using hyperspectral images. The method proposed spatial-spectral GATs (**S^{2}GATs**) to apply the GATs in their domain (Refer Fig. 1). At the front, K-nearest neighbor is performed through spatial-spectral Euclidean distance measurement. After that, the graph model is used to assign specific values to describe the connection strength to other neighbors. In **S^{2}GATs**, important neighbors are emphasized in the final summation by measuring similarity according to each node through the attention mechanism during graph convolution and generating weights. After that, semi-supervised classification is performed through the FC layer. By assigning different weights to different neighbors during graph convolution process, it could avoid artificially designing connection weights like vanilla graph convolution. Additionally, the reason that the method presented above achieved the best accuracy is that distance measurement and importance are effectively learned by the attention mechanism.

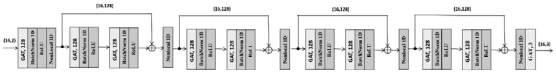


Figure 2. The architecture of the Graph Attention Residual Networks (**GARNet**)

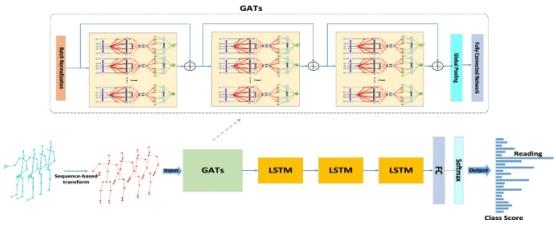


Figure 3. The architecture of the view transform graph attention recurrent networks (**VT + GARN**)

B. *Pose estimation*

Graph attention residual networks (**GARNet**) [3], a network using GATs, is proposed to deal with the adversarial learning framework for 3D Human Pose Estimation. **GARNet** learns the mapping from 2D key points distribution to 3D keypoints distribution. The structure of **GARNet** is shown in denoted in the Fig. 2.

GARNet is composed of GATs and non-local layers as follows, in order to obtain the relationship between local and non-local layer nodes. In addition, after performing GATs two times, the residual block is formed in a continuous manner. And the discriminator is trained to correctly guess whether the 3D pose is the ground truth or not, causing the generator to generate a 3D pose that more closely resembles the ground truth. **GARNet** apply GATs, not vanilla GCN, as a way to process graph neural networks, due to the fact that when GCN are trained based on one domain, this network is not sufficient to show similar performance in other domains.

C. *Action recognition*

Skeletons are useful for revealing the actions of the human bodies; while they inherently in the non-Euclidean space. In [7], a model called as view transform graph attention recurrent networks (**VT + GARN**) is proposed to effectively tackle the complicated representation of spatial-temporal skeleton sequences when a large change in action obtained from different viewpoints is given. It converts the original data into view-invariant data through view-transform, puts it as an input to the graph attention networks, extracts spatial features, and then executes classification tasks by expressing temporal features using LSTM. The structure of **VT + GARN** is shown in Fig. 3.

Within the model, GATs calculates attention coefficients for each neighbor node and extracts spatial features from the skeleton data. The reason for using GATs instead of using GCN in the model is to assign different weights

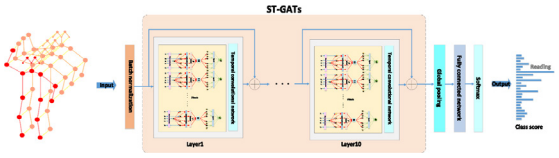


Figure 4. The architecture of the spatial-temporal graph attention networks (**ST – GATs**)

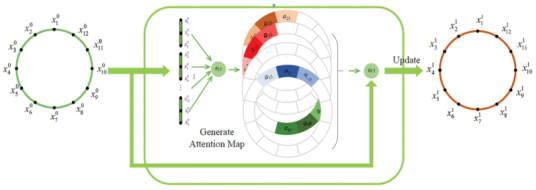


Figure 5. The GAT module

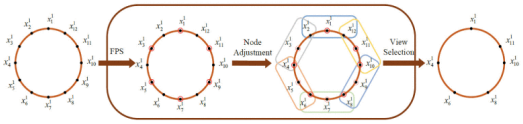


Figure 6. The view selection module

to joint nodes using self-attention method since there is a problem that GCN uses the same weight even though the influence of each node on motion is different.

Another model for skeleton-based action recognition, spatial-temporal graph attention networks (**ST – GATs**) [6], is also introduced. Unlike VT+GARN described above, this constructs a spatiotemporal graph using the skeleton sequence observed only in one view, and then passes it through the **ST – GATs** multi-layer network. The ST-GAT module determines the spatiotemporal neighbors of the root node according to the neighbor function and calculates the attention coefficients for each neighbor node. Finally, a class score is obtained using features and attention coefficients of neighbor nodes. To construct the **ST – GATs**, a spatiotemporal graph attention layer was added to the TCN [9] framework. This network includes 10 layers of spatial-temporal graph attention operators. And since multi-head attention can play a role in preventing over-fitting of the network including multiple-independent attention for same input, K-independent head attention is applied. GATs updated the node features of the skeleton in this network to show better performance. And Figure 4 describes structure of **ST – GATs**.

According to the paper, vanilla GCN greatly improved the accuracy of skeletal action recognition, but vanilla GCN has the drawback that graph convolution uses the same weight for all nodes and is highly dependent on graph structure. Therefore, they use GATs which successfully integrate the benefits of graph convolution and self attention mechanism, and contribute to improved performance. Also compared with TCN used as a baseline, the biggest difference from TCN is that different weights are assigned to each neighbor node, this shows a huge difference in the experimental results. And through comparison with another baseline, ST-GCN [16], it is found that multi-head attention is contributing to the performance.

This time, we will introduce a network using GATs for 3D object recognition. It is important to make the most of the information available from multiple views for 3D object recognition. To this end, hierarchical graph attention based multi-view convolutional neural network (**GA – MVCNN**) [18] is proposed. This network is divided into three main steps. The first is the process of projecting a 3D object as a 2D image series for view feature extraction and putting it as an input to the CNN. The second is the process of selectively aggregating features from multiple view to generate global features through the view graph attention network module and correlation weighted feature aggregation. Lastly, 3D object recognition using global features. Next, we would like to investigate the role of GATs in **GA – MVCNN** presented above. GATs is used for view selection in the second process of the main stage. For this, the view feature from view feature extraction is defined as a node, and the adjacent projective positional relationship between views is defined as an edge. And update the nodes of this graph by using GATs, select and remove duplicated views through the view selection module. Figure 5 and Figure 6 are illustrations of the above process.

In [18], the reason for using GATs to deal with the graph structure is described, and the reason is as follows: First, since graph convolution in the frequency domain depends on the Laplacian matrix and is affected by the graph structure, a model learned from a specific graph cannot be directly applied to other graphs. Second, the improvement of the performance of 3D object recognition by giving more weight to the views related to the attention mechanism is presented as an advantage. In addition, an experiment is conducted to remove the GAT module and confirmed that the performance has deteriorated. It can be seen that GATs is contributing to the performance of the **GA – MVCNN** method.

D. Few-Shot Learning

We will explain the graph attention propagation network (**GAP**) [8], a network that uses GATs for Few-Shot Learning, which learns a new category with small amount of data from a base categories with sufficient training samples. It uses a method of learning new categories using correlation between base categories and novel categories. **GAP** uses graph attention mechanism to find correlation between categories and uses this to deliver information. This method repeatedly initializes the parameters of all categories and

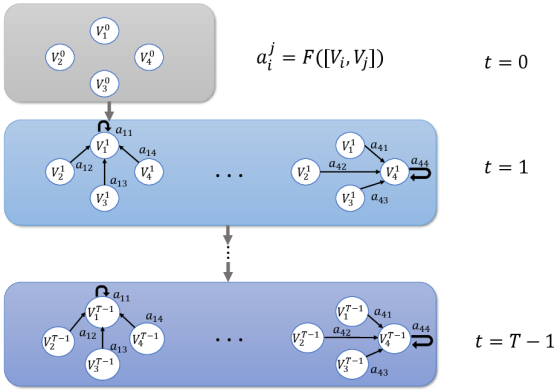


Figure 7. The graph attention propagation networks

updates the parameters by calculating the correlation. In this way, the model can use the information about base categories to learn about the new categories, and Figure 7 describes how it works.

Additionally, there is a problem that computation is expensive and most nodes are not related to calculating the correlation for all nodes for each node. Therefore, reasonable constraints are needed to solve this problem. In [8], semantic information is used for control the attention space. It is a method to obtain a semantic vector for a category by calculating the euler distance for other nodes, and use only the closest ones.

E. Visual tracking

Target-aware simaese graph attention tracking (**SiamGAT**) [5], a network for tracking, uses a graph attention module (GAM) inspired by GATs. This is proposed for part-to-part matching for information embedding, and compared with the existing cross-correlation-based method, the disadvantages are greatly eliminated and target information can be efficiently transmitted from the template to the search region. More specifically, the existing method ignores the part-level correspondence between the target and the search region, so matching for the shape and posture of the target is not accurate. In addition, the method of delivering target information by compressing it too much can cause information loss. To solve this problem, [5] define part-to-part correspondence between the target template and the search region using a complete bipartite graph. This creates a bipartite graph using the target-aware tracking template-feature-area selection mechanism on the feature maps F_t and F_s obtained after passing the template patch T and search region S through the Siamese feature extraction network. After obtaining the correlation score between each node using this, a response map for subsequent work is generated through a process similar to the operation presented in the existing GATs to update the node feature. An overview of

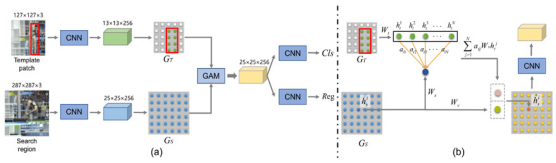


Figure 8. (a) The Simaese Graph Attention Tracking network and (b) Graph Attention module

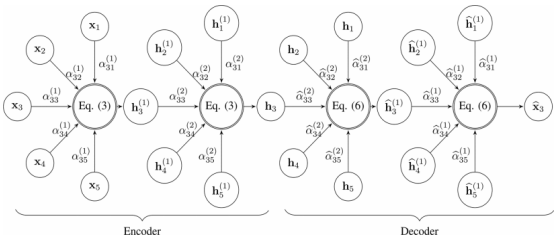


Figure 9. The 2-layer Graph Attention Auto-Encoder networks

the model is in Figure 8. (a) is the network architecture for **SiamGAT**, and (b) describes the graph attention module. Additionally, it can be seen from the ablation study that GAM enabled target-aware feature area selecting, the method presented in this paper, and greatly improved tracking performance.

F. Image generation

A graph auto-encoder structure is proposed to take advantage of the relation information of data in the graph structure. In previous work, there was a problem that there was no interest in reconstructing graph structure or node attribute. To solve this problem, graph attention auto-encoder (**GATE**) [11] was introduced, a deep learning model for unsupervised representation learning on graph structure data. Figure 9 is an illustration of a 2-layer graph attention auto-encoder for reconstructing graph node features.

First, the encoder receives node features as an input and generates a node representation using the graph structure. Each encoder layer uses its relationship with its neighbors to create a new representation of the node, just like GATs did. The decoder reverses the encoding process, and the number of layers of the decoder is the same as the number of layers of the encoder. At this time, each decoder layer reconstructs the node representation by using the relationship with the neighbor. Finally, to reconstruct the graph structure and node features, two types of loss are used to minimize the reconstruction loss. For node features, L_2 loss is used, and for graph structure a loss that makes representations of neighboring nodes similar is used. Additionally, **GATE**'s operation is very efficient since it can process operations including graph attention mechanism in parallel. And the complexity of time that can be known through theoretical analysis can tell us the fact that this model is theoretically efficient.

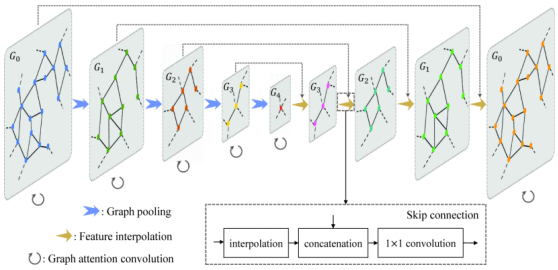


Figure 10. The Graph Attention Convolution Network (**GACNet**)

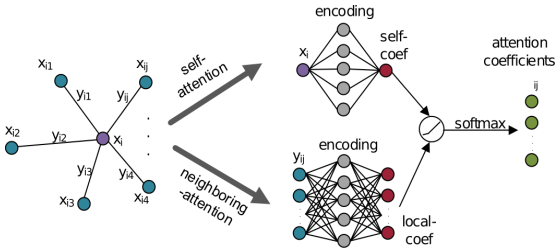


Figure 11. The Graph Attention based Point Neural Network (**GAPNet**)

G. 3D point cloud data

In the last section, we are going to cover two networks that deal with Point Cloud data. First, we will introduce novel graph attention convolution network (**GACNet**) [15]. Since vanilla GCN uses a spectral representation of a graph, computation is too expensive, and a spectral GCN model trained on one graph cannot be transferred to a graph with another Laplacian matrix. In order to solve the shortcomings of vanilla GCN, this network is inspired by GATs that integrate non-spectral method and attention mechanism. In GAC, the convolution kernel of GAC dynamically accommodates the object structure through the sharing attention mechanism, a method that pays attention only to the most relevant part according to the feature attribute of the neighbor. The input of GAC becomes a vertex feature set, and the output becomes an updated vertex feature set. Based on this structure, a graph attention convolution network (**GACNet**) is created, a network for point cloud semantic segmentation. In the graph pyramid, GAC is applied to learn local features in each layer, and the resolution is reduced through graph pooling. After that, the features learned from the graph pyramid are interpolated to obtain a feature map with the same number of points as the original input. Finally, an additional GAC layer is applied in consideration of the loss due to graph pooling or feature interpolation layer.

Another network for handling point cloud data, graph attention based point neural network (**GAPNet**) [2], will be introduced. It is inspired by GATs and mainly focused on fully exploiting local features for point clouds in an attention manner. The proposed model for better learning local representations of unstructured point clouds in classification and segmentation tasks consists of three types: GAP Layer, attention pooling layer, and **GAPNet** architecture. Additionally, the model uses only 3D coordinates as input features. In GAP Layer, a k-nearest neighbor graph is constructed since the point cloud is too large for practical applications, and if all points are involved in all other points, the computation cost and gradient vanishing problem due to very small weight assignment occur. Then, attention coefficients for each neighbor pair are generated by fusing the coefficients obtained using self-attention and neighboring-attention. Its structure is as follows:

Here, self-attention considers self-geometric information for each point, and neighboring-attention focuses on local-coefficients by considering neighborhood. By using attention coefficients obtained through the above structure, attention feature is taken as an output. In addition, M independent single-head GAP Layers are concatenated to obtain sufficient structural information and network stability, and an attention pooling layer is used to enhance the robustness and performance of the network. Finally, a **GAPNet** architecture that combines spatial transform and MLP layer for classification and segmentation tasks is presented. Through the ablation study in [2], it can be seen that the graph attention mechanism is having an effect on the performance by seeing that the accuracy increases when GAP Layer is used. And the good results of the model prove the efficiency and geometric relationship understanding of the graph attention networks.

V. Conclusion

In this paper, we have investigated various computer vision applications dealing with graph structured data using GATs. First, Vanilla GCN, GATs, and methods that improved GATs are investigated. Then, through the comparison between Vanilla GCN and GATs, we summarized the pros and cons of the architectures (i.e. Vanilla GCN, GATs) for the specfic computer vision tasks in the later sections.

VI. Acknowledgement

This work was supported by the IITP grants (No. 2021–0–01778, Development of human image synthesis and discrimination technology below the perceptual threshold) funded by the Korea government (MSIT).

References

[1] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. In *ICLR*, 2014.

[2] Can Chen, Luca Zanotti Fragonara, and Antonios Tsourdos. Gapnet: Graph attention based point neural network

for exploiting local feature of point cloud. In *arXiv preprint arXiv:1905.08705*, 2019.

[3] Zhihua Chen, Xiaoli Liu, Bing Sheng, and Ping Li. Garnet: Graph attention residual networks based on adversarial learning for 3d human pose estimation. In *Computer Graphics International Conference*, pages 276–287. Springer, 2020.

[4] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, volume 29, pages 3844–3852, 2016.

[5] Dongyan Guo, Yanyan Shao, Ying Cui, Zhenhua Wang, Liyan Zhang, and Chunhua Shen. Graph attention tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9543–9552, 2021.

[6] Qingqing Huang, Fengyu Zhou, Jiakai He, Yang Zhao, and Runze Qin. Spatial–temporal graph attention networks for skeleton-based action recognition. In *Journal of Electronic Imaging*, volume 29, page 053003. International Society for Optics and Photonics, 2020.

[7] Qingqing Huang, Fengyu Zhou, Runze Qin, et al. View transform graph attention recurrent networks for skeleton-based action recognition. In *Signal, Image and Video Processing*, volume 15, pages 599–606. Springer, 2021.

[8] Xiaolu Hui, Riquan Chen, and Tianshui Chen. Graph attention propagation for few-shot learning. In *Proceedings of the ACM Turing Celebration Conference-China*, pages 1–4, 2019.

[9] Tae Soo Kim and Austin Reiter. Interpretable 3d human action analysis with temporal convolutional networks. In *2017 IEEE conference on computer vision and pattern recognition workshops (CVPRW)*, pages 1623–1631. IEEE, 2017.

[10] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[11] Amin Salehi and Hasan Davulcu. Graph attention auto-encoders. In *arXiv preprint arXiv:1905.10715*, 2019.

[12] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. In *IEEE transactions on neural networks*, volume 20, pages 61–80. IEEE, 2008.

[13] Anshu Sha, Bin Wang, Xiaofeng Wu, and Liming Zhang. Semisupervised classification for hyperspectral images using graph attention networks. In *IEEE Geoscience and Remote Sensing Letters*, volume 18, pages 157–161. IEEE, 2020.

[14] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018.

[15] Lei Wang, Yuchun Huang, Yaolin Hou, Shenman Zhang, and Jie Shan. Graph attention convolution for point cloud semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10296–10305, 2019.

[16] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

[17] Yang Ye and Shihao Ji. Sparse graph attention networks. In *IEEE Transactions on Knowledge and Data Engineering*. IEEE, 2021.

[18] Hui Zeng, Tianmeng Zhao, Ruting Cheng, Fuzhou Wang, and Jiwei Liu. Hierarchical graph attention based multi-view convolutional neural network for 3d object recognition. In *IEEE Access*, volume 9, pages 33323–33335. IEEE, 2021.

[19] Jiani Zhang, Xingjian Shi, Junyuan Xie, Hao Ma, Irwin King, and Dit-Yan Yeung. Gaan: Gated attention networks for learning on large and spatiotemporal graphs. In *UAI*, 2018.

A Survey on Recent 3D Hand Pose Datasets

Hansoo Park[1], Yunyoung Jeong[2], Yunhoe Ku[2], Seungeun Lee[2] and Seungryul Baek[1]

[1] UNIST, AI Graduate School, Ulsan, South Korea, {hansupark, srbaek}@ unist.ac.kr
[2] UNIST, Computer Science and Engineering, Ulsan, South Korea, {kks6716, selee, myhjyy}@unist.ac.kr

Abstract

Three-dimensional hand pose estimation is one of the main tasks in the computer vision field. While deep learning-based approaches that require many datasets are prevail in this task; creating 3D hand pose datasets takes a lot of manual efforts. Especially, the 3D labeling procedure becomes increasingly difficult when we meet the image data with occlusions. Recently, several researchers proposed methods for obtaining accurate annotations with realistic images. In this paper, we summarized the recent 3D hand pose estimation benchmarks within 5 years.

Keywords— *3D Hand pose estimation, Computer vision, Deep learning*

I. Introduction

In the real world, the human hand is one of the parts that human often use when manipulating something. Therefore, in order to understand human behavior, it is important to recognize the hand pose. In computer vision, the hand pose data consists of 63 skeletons and 21 joints. And 3D Hand pose estimation is a problem that predicts the position and depth of each joint from the image and minimizes the distance between ground-truth and predicted result.

Various methods have been coming out to solve the problem. Recently, as it is known that hand pose estimation performance can be improved through deep learning. So, many researchers focus on the deep learning-based method and require datasets which is a key requirement of the deep learning model.

However, it is not easy to create the hand pose dataset. Due to factors such as viewpoint and articulation, some joints may become occlusion, which may reduce annotation quality. The problem of predicting the posture of the hand in a situation where the hand and an object or other hand interacts is drawing attention. Since these tasks generate more complex occlusions, the labeling process has become more complicated. In addition, it is also important to expand the variation of datasets since the hand have many factors to change the appearance of hand such as hand shape, view point change and many degrees of freedom(DoFs). We will explain more details on that challenges in Section.ii.. In this paper, we will introduce 3D Hand pose estimation benchmark datasets. Also, we will describe the method used to solve the above problems in each dataset.

II. Challenges in creating dataset

In order to solve the 3D hand pose estimation problem through deep learning, the dataset containing large amounts of training data with all possible variations in each factor of hand appearance is necessary. [11] However, creating the hand pose dataset takes a lot of resources and time due to many challenges. In this section, we describe two of the main challenges, annotation quality and variation, in detail and how to solve them.

A. Annotation quality

In order to maintain the performance of the model during testing, it is important that the dataset has highly accurate annotations. The best way to get highly accurate annotations is to label them directly by humans. However, since it requires a lot of resources to label them directly, it is impossible to obtain sufficient data. Especially, Hand pose estimation requires annotation of 21 joints and unlike the face or body, many occlusion occurs. The annotator needs to estimate and annotate them directly when occlusion occurs, So, it takes a lot of time and lowers annotation accuracy. [19] Therefore, many researchers have studied various methods that can improve annotation accuracy while minimizing human intervention. Typically, there are methods using a device such as sensors [26, 4, 1, 22], multiple cameras [13, 19, 30, 3] and a model-based method [5, 8, 30, 3].

B. Variation

The human hand can have various hand poses depending on the hand shape, view point change and degrees of freedom(DoFs). So, it is important to create the hand pose dataset with all possible variations in each factor.

We can expand the variation of dataset through increasing the number of subjects to capture various hands or using multiple cameras to obtain appearances that depend on the viewpoint. However, these methods require a lot of resources. When there exists the resource limitation, the method of creating synthetic images can be used. It is relatively free from resource constraints since it can create various hands by using model parameter values and render hand images from multiple viewpoints.

III. OVERVIEW

In the previous section, we reviewed the challenges that occur when creating the hand pose dataset. In this section, we introduce the criteria that can separate datasets. Also, we introduce 3D hand pose estimation benchmark datasets separately before and after 2019.

A. *Criteria*

The dataset used for hand pose estimation can be divided with various criteria. (e.g. visual modality, data type, existence of object or other hand in images, labeling method) In this subsection, we describe the visual modality, data type and labeling method that should be prioritized when creating or selecting the dataset.

1) *Visual modality*

The 3D hand pose estimation method mainly uses data in Depth, RGB, and RGB+Depth format. Depth and RGB+Depth images have depth information, so it provides strong interpretability, but the price of related equipment is high and there are many restrictions on the implementation. On the contrary, RGB images have lower requirements for equipment and implementation, but provide lower interpretability than the above data formats. [10]

2) *Data type*

Real: As the method using real images as the training dataset, [26, 4, 1, 13, 19, 30, 5, 3] belong to this type. Since realistic features can be obtained from real images, the performance of the model can be maintained in real scenes. However, getting annotation information from real images requires a lot of resources, so it may contain very poor quality annotations to get enough data. Also, since the number of subjects is limited, it may be limited to include various hands. [11]
Synthetic: As the method generating synthetic images by specific methods and using them as training data, [29, 14, 6, 15] belong to this type. While creating images, annotations are also generated automatically. So, accurate annotation information can be obtained with relatively few resources. In addition, by giving values to the parameter of the method, various hand appearances can be included in the dataset. However, the domain gap between real and synthetic can reduce the performance of the model in real scene.

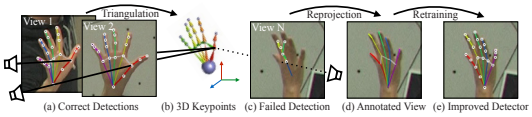


Fig. 1. Multiview Bootstrapping. (a) A multiview system provides views of the hand. Some views which detect hand correctly are used to triangulate. (b) is the 3D keypoints generated by triangulation. Views that failed detections as (c) can be annotated using the reprojected 3D keypoints (d), and used to retrain an improved detector (e).

3) *Labeling*

Marker-based: As method attaching devices to the hand and tracking them to get the annotation information of the hand, [26, 4, 1, 22] use this method. In this method, the position and rotation of the joint can be measured without prior knowledge of the hand skeleton, so that accurate annotation information can be obtained efficiently. However, appearance changes and unnatural hand motion may occur due to devices attached to the hand. [3]
Marker-less: As the method using only images to obtain hand annotation information, [26, 4, 1, 22] use this method. This uses hands with nothing attached, so we can get the natural hand motion. However, the additional logic should be required, since this method uses only images to create annotation information. [5, 8] utilize the hand model generating the hand mesh and optimize the model by comparing 3D keypoints generated from mesh and ground-truth. [19, 13, 30, 3] utilize multiple cameras to generate 3D annotations.

B. *Dataset before 2019*

In this subsection, we introduce 3D hand pose estimation benchmark datasets released before 2019. We mainly introduce how to create annotations and expand variation in each dataset. In addition, information about datasets such as the number of frame, subject, object and resolution is shown in the Table.1.

1) *Panoptic studio* [7]

Panoptic studio [7] is a dataset that captures 2D hand poses in the multi-view environment. [19] introduced the method estimating 2D hand keypoints and automatically generating 3D keypoints using this dataset and multi-view geometry. This process is shown in Fig. 1.

The performance of the 2D hand keypoint detector was improved by repeating the above process. After training, the 3D keypoints could be generated by triangulating the 2D keypoints of the multi-view via the calibration matrix.

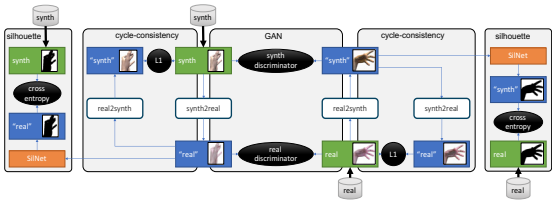


Fig. 2. Network architecture of GeoConGAN. The trainable part comprises the real2synth and the synth2real generator and discriminator. The loss functions are shown in black, real images in green, synthetic images in blue, and the SilNet in orange.

Through this, more accurate 2D keypoints and 3D keypoints could be obtained by above process and the improved 2D keypoint detector.

2) BigHand2.2M [26]

The BigHand 2.2M dataset was created for single hand pose estimation. This dataset used magnetic sensors to obtain annotations. After acquiring the location and rotation of some joints (5 finger tips, 1 back of palm) with sensors, this dataset used inverse kinematics to get information about other joints. Since then, the ASPnp algorithm [27] was used to integrate the 3D location obtained from the magnetic sensor and the 2D location obtained from the depth camera into a single system.

The BigHand2.2M dataset included images on multiple viewpoints by setting the height of the sensor, the position of the subject and the orientation of the arm to widen the variation of the dataset. In addition, the process of moving from a specific pose to another pose was captured. And it expanded the pose space by including random poses and images from egocentric view.

3) FPHA [4]

The FPHA dataset was created for the problem of recognizing the action of hands that interact with an object in an egocentric view. Labeling process was done by utilizing magnetic sensors and inverse kinematic like [26]. In addition, the additional sensor was attached to the part closest to the center of the object that can be reached to obtain the object pose.

By not providing information about speed or style of action to subject, the FPHA dataset could include realistic and broader range of actions.

4) GANerated Hands [14]

The GANerated Hands dataset was created for the 3D hand tracking problem and used synthetic images. This dataset proposed the GeoConGAN algorithm to translates a synthetic image into a real image based on CycleGAN [28] using unpaired datasets.

This dataset added geometric consistency term to GeoConGAN for maintaining the hand pose shape between synthetic and real images. The network architecture of GeoConGAN is shown in Fig. 2.

Geometric consistency term used the cross entropy loss function with the silhouettes of the two images, real and synthetic image. At this time, SilNet based on UNet [17] was used to generate the silhouette.

The GANerated Hands dataset expanded the variation of the dataset by mixing images on the egocentric view of SynthHands [15] and images on the 3rd view obtained via the hand tracker or hand animation platform.

This dataset tried to solve the domain gap, which was the biggest drawback of the synthetic image, via GeoConGAN. It could also maintain annotation quality without the need for additional annotation methods by creating real images via synthetic images.

5) Rendered Hand Pose Dataset(RHD) [29]

The RHD dataset was created for single hand pose estimation and used synthetic images. This dataset generated 3D hands from Mixamo [1] and rendered them using Blender software.

To widen the variation of the dataset, it randomly set the intensity and position of the lights and the shape of the hand. They also included various hand appearances changed by using multiple viewpoints which could see a part of the hand at least.

6) SynthHands [15]

The SynthHands dataset was created for hand-object interaction and used synthetic images. This dataset used the hand motion captured in the real world and the virtual object to generate new data. The hand annotations were automatically annotated through real-time markerless tracker [21]. After this, they were re-targeted to the synthetic images by artists.

The SynthHands dataset included various hand poses by setting randomly skin, gender and hand shape and rendering them. Also, they set various values of location, rotation and scale and applied various textures and shadings to objects with chroma keying.

The SynthHands dataset created synthetic images based on the actual hand motion by the tracker. Through this, it could overcome the drawback of the synthetic image that it can have the unnatural hand motion. Also, it could generalize unseen objects by not using the object model and scan.

C. Dataset after 2019

In this subsection, we introduce 3D hand pose estimation benchmark datasets released after 2019. We describe

[1] http://www.mixamo.com

this section in the same way as section B.. The additional information about these datasets is shown in the Table. 1.

1) HO-3D [5]

The HO-3D dataset was created for hand-object interaction task. The optimum hand pose and object pose were obtained by using the energy function. This dataset obtained 3D hand poses and 3D object poses via the MANO hand model [16] and YCB-video [25], respectively. It also got ground-truth by putting 2D images on multi-view into the model. After that, they were inserted into the energy function to get the optimum hand pose and object pose. The authors represented the energy function as follows:

$$\hat{\mathscr{P}} = \arg\min_{\mathscr{P}} \sum_{t=1}^{N_F} \left(E_{\mathscr{D}}(\mathrm{p}_{\mathrm{h}}^t, \mathrm{p}_{\mathrm{o}}^t) + E_{\mathscr{C}}(\mathrm{p}_{\mathrm{h}}^t, \mathrm{p}_{\mathrm{o}}^t) \right), \qquad (1)$$

where E_{D}, E_{C}, p_{h}, p_{o} and $\mathscr{P}$ represent the data term energy, constraints energy, hand pose, object pose and a set including both object pose and hand pose.

The data term energy was used to minimize the distance between predicted result and ground-truth with segmentation mask, depth and 2D joint. The constraint energy was used to prevent from generating impossible hand pose and maintain temporal consistency. By minimizing the energy function, the optimum 3D hand pose and object pose were obtained.

This energy function reduced the impact of falsely estimated ground-truth obtained by the model through the constraint term. As a result, the HO-3D dataset could reduce time resources and improve the quality of annotations.

2) DexYCB [3]

The DexYCB dataset was created for hand-object interaction. The optimum hand pose and object pose were obtained by using the energy function. This dataset obtained 3D hand poses and 3D object poses via the MANO hand model and YCB-video as HO-3D dataset. Also, in order to obtain the ground-truth keypoint for evaluating predicted poses, the annotators marked keypoints to the 2D images on multi-view in advance. After that, they used the energy function to get the optimum hand pose and object pose with predicted poses and ground-truth. The authors represented the energy function as follows:

$$E(P) = E_{\mathrm{depth}}(P) + E_{\mathrm{kpt}}(P) + E_{\mathrm{reg}}(P), \qquad (2)$$

where P, E_{depth}, E_{kpt} and E_{reg} represent a set including a hand pose and object pose, depth error term, reprojection error term and regularization term, respectively. E_{depth} measured how well the pose represents the observed depth data through the calculation of whether the points in the ground-truth point cloud exist the surface of the predicted mesh. E_{kpt} measured the error by calculating the distance between the ground-truth 2D keypoints and predicted 2D keypoints by reprojecting the 3D keypoint to 2D multi-view. E_{reg} was the regularization term for the MANO parameter. By minimizing the energy function, the optimum 3D hand pose and object pose were obtained.

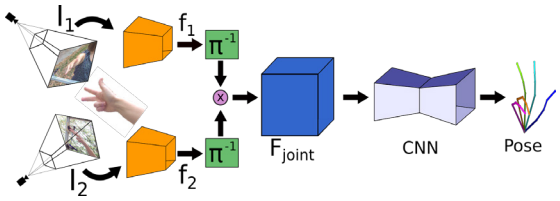


Fig. 3. MVNet architecture: MVNet predicts a 3D hand pose using images of multiple views. Each feature map f_i is generated by passing through a 2D CNN that is shared across views. These are individually reprojected into a common coordinate frame F_i using the known camera calibration Π^{-1}. F_{joint} is generated by aggregating all F_i and finally passes through 3D CNN to localize 3D keypoints.

The authors asked each subject to choose 2-4 objects and then captured the overall process from relaxed state to grasping the object and holding. Through it, this dataset could expand the variation.

For Dexycb dataset, the accurate annotations could be obtained by manually annotating for images on multi-view. Thanks to this, it was able to accurately evaluate the predicted 3D keypoint.

3) FreiHAND [30]

The FreiHAND dataset was created for 3D hand pose estimation. This dataset obtained the 3D hand pose via the MANO hand model and ground-truth 2D keypoint on multi-view obtained by manually labeling. Instead of labeling all 21 keypoints, the authors annotated only 6 keypoints(5 fingertips, 1 wrist). Since then, 3D keypoints were generated by putting the keypoints of several views into MVnet. The architecture of MVNet is shown in Fig 3.

After that, they used the energy function to get the optimum hand pose with predicted poses and ground-truth. The authors represented the energy function as follows:

$$\mathscr{L} = \mathscr{L}_{\mathrm{kp}}^{\mathrm{2D}} + \mathscr{L}_{\mathrm{kp}}^{\mathrm{3D}} + \mathscr{L}_{\mathrm{seg}} + \mathscr{L}_{\mathrm{shape}} + \mathscr{L}_{\mathrm{pose}}. \qquad (3)$$

$L_{\mathrm{kp}}^{\mathrm{2D}}$, $L_{\mathrm{kp}}^{\mathrm{3D}}$ and L_{seg} represented the distance value between 2D keypoint, 3D keypoint and segmentation mask of ground-truth and predicted result. L_{shape} was the term that regularizes the generated shape so that it is close to the mean shape of MANO. L_{pose} was the term makes the predicted pose close to the surrounding poses in terms of MANO's PCA pose space. The MANO hand model was optimized through above multi loss term. After that, the verification was performed on the results obtained through the MANO hand model and MVNet, and a part of the rejected data was used as retraining data.

This dataset could improve the annotation quality by using 3D annotation as ground-truth when fitting the MANO hand model with multi-view system and MVNet.

4) InterHAND2.6M [13]

The InterHAND2.6M dataset was created for hand-hand interaction. The process of creating dataset was similar to [19]. However, when annotating some images in order to train the detector, unlike which obtained annotations by utilizing the model at the beginning, this dataset obtained them by manually labelling. The authors used the annotation tool which generates automatically annotations for other 6 views if the annotator marks on only 2 views per frame. Also, [19] repeatedly bootstrapped the initial detector but the InterHAND2.6M dataset trained the keypoint detector using the EfficientNet [23] as the backbone only once. Since the InterHAND2.6M dataset could use sufficient data for training during the above process, it was possible to train only once.

The InterHAND2.6M dataset contained single hand images and hand-hand interaction images. It also included a wide variety of possible hand poses by including the overall process that shifts from a specific pose to the other pose and a hand-hand interacting movement with gestures during conversation.

5) YouTube 3D Hands [8]

The YouTube 3D Hands dataset was created for 3D mesh reconstruction and pose estimation for single hand. There was a drawback that most images in other datasets were captured in the laboratory, So, the captured data in other environments was not generalized. This dataset tried to solve this problem by collecting data from Youtube including images in various domain.

In this paper, the MANO model was iteratively fitted to optimize the procedure of lifting the 2D keypoint obtained by OpenPose to the 3D Mesh. After that, the 3D mesh was converted into the 3D keypoint. The process of converting the 3D mesh to 3D keypoints is expressed as follows:

$$J(\beta, w, \vec{T}_\delta, s) = \mathscr{J}^T M(\beta, P(w), \vec{T}_\delta, s). \quad (4)$$

The equation 4 was used to find the optimal matrix $\mathscr{J}^T$ converting the 3D mesh M into the 3D keypoints J. The process of finding the optimal shape β and rotation w is expressed as follows:

$$\{\beta^*, w^*, \vec{T}_\delta^*, s^*\} = \arg\min_{\beta, w, \vec{T}_\delta, s} (E_{2D} + E_{bone} + E_{reg}), \quad (5)$$

where E_{2D}, E_{bone} and E_{reg} represent reprojection error term, bone loss term and MANO parameter regularization term, respectively.

The authors combined images generated in above way with hand images of the COCO dataset [9]. Since this dataset included data from various domains on Youtube, the variation of the dataset could be expanded.

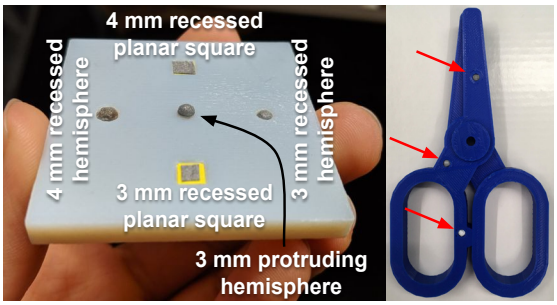


Fig. 4. **Left**: Different object tracking marker configurations. **Right**: 3D printed object and recessed 3 mm hemispherical markers(highlighted by red arrows)

6) ContactPose [1]

The ContactPose dataset was created for hand-object interaction task. This dataset provides hand poses, object poses and contact maps, which are thermal image on the object.

The ContactPose dataset obtained the 3D hand joint location via multi-view and the object pose using a motion capture system. This dataset used a small number of markers to get the accurate object pose. In addition, the surface of the marker was deformed to make it visible even in situations where occlusion occurred due to various angles and grips. The configuration proposed in this paper is shown Fig. 4. The optimum 3D keypoint was obtained based on the hand pose and object pose obtained in above way. It was expressed as follows:

$$\min_{^{o}\mathbf{X}} \sum_{i=1}^{N} \sum_{c=1}^{C} \mathscr{D}\left(\mathbf{x}_c^{(i)}, \pi\left({}^{o}\mathbf{X}; K_c, {}^{c}T_o^{(i)}\right); \mathbf{w}_c^{(i)}\right) \quad (6)$$

After projecting the 3D keypoint to 2D with preojection function π, considering object pose ${}^{c}T_o^{(i)}$ and camera intrinsics K_c, the optimal 3D keypoint was obtained by minimizing error calculated by distance function $\mathscr{D}$ for the reprojected 3D keypoint on each view and the corresponding 2D keypoint.

Also, the optimum hand mesh was obtained by comparing the 3D keypoint obtained in above way and the 3D keypoint generated from the 3D hand mesh.

Since this dataset used multiple cameras to obtain 3D joints, the quality of annotation did not deteriorate even if occlusion occurs due to the object. In addition, since hand-object pose and contact information were provided together, more interaction information could be utilized in hand-object interaction.

7) ObMan [6]

The ObMan dataset was created for hand-object interaction task and used synthetic images. This dataset captured the hand on a full body hand using the SMPL+H [16] to get the natural hand. Then, it used the object model provided by ShapeNet [2] to generate the object. The objects and hands obtained in this way were used to generate grasps using the software GraspIt[12].

To widen the variation of dataset, the ObMan dataset used the SURREAL [24] providing various body pose values and randomly selected the body rotation and camera distance. It also used images in LSUN [20] and ImageNet [18] dataset as background to reduce gaps between created images and real images.

By creating and using synthetic images used for hand-object interaction, it was possible to solve the problem of annotation quality deterioration due to occlusion caused by object. Also, by generating grasps via the software using the robotics algorithm, it was possible to include realistic hand-object interaction images.

IV. Conclusion

We introduced the 3D hand pose estimation benchmark datasets and explained how each dataset solved the challenges that arise while creating the hand pose dataset. Datasets can be broadly divided according to type of image, which can be real or synthetic. For real image, the methodology for improving annotation quality has been studied, and for synthetic image, the methodology for reducing the gap with the real world has been studied.

Real images are separated according to whether to use markers, and the datasets without marker use additional methods such as the MANO hand model [16] and multi-view to improve annotation quality. Synthetic images either create hands through software, hand model or capturing the actual hand motion [15].

We also split 3D hand pose datasets before and after 2019 and then summarize them. While summarizing them, we could find out that most of the real image-based datasets after 2019 used the objective function and the MANO hand model to obtain the optimum annotation.

Both real type and synthetic type still have its limitations. In particular, the hand-object interaction task and hand-hand interaction task, which have been attracting attention recently, require the more complicated annotation mechanism since not only self-occlusion but also occlusion by other elements occurs in these tasks. Therefore, we need to do more research on more accurate annotation methods and ways to create more realistic hands than current methods.

V. Acknowledgement

This work was supported by the IITP grants (No. 2021–0–01778, Development of human image synthesis and discrimination technology below the perceptual threshold) funded by the Korea government (MSIT).

References

[1] Samarth Brahmbhatt, Chengcheng Tang, Christopher D Twigg, Charles C Kemp, and James Hays. Contactpose: A dataset of grasps with object contact and hand pose. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*, pages 361–378. Springer, 2020.

[2] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[3] Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, et al. Dexycb: A benchmark for capturing hand grasping of objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9044–9053, 2021.

[4] Guillermo Garcia-Hernando, Shanxin Yuan, Seungryul Baek, and Tae-Kyun Kim. First-person hand action benchmark with rgb-d videos and 3d hand pose annotations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 409–419, 2018.

[5] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Honnotate: A method for 3d annotation of hand and object poses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3196–3206, 2020.

[6] Yana Hasson, Gul Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning joint reconstruction of hands and manipulated objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11807–11816, 2019.

[7] Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social motion capture. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3334–3342, 2015.

[8] Dominik Kulon, Riza Alp Guler, Iasonas Kokkinos, Michael M Bronstein, and Stefanos Zafeiriou. Weakly-supervised mesh-convolutional hand reconstruction in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4990–5000, 2020.

[9] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[10] Yang Liu, Jie Jiang, and Jiahao Sun. Hand pose estimation from rgb images based on deep learning: A survey. In *2021 IEEE 7th International Conference on Virtual Reality (ICVR)*, pages 82–89. IEEE, 2021.

[11] Jameel Malik, Ahmed Elhayek, Fabrizio Nunnari, Kiran Varanasi, Kiarash Tamaddon, Alexis Heloir, and Didier Stricker. Deephps: End-to-end estimation of 3d hand pose and shape by learning from synthetic depth. In *2018 International Conference on 3D Vision (3DV)*, pages 110–119. IEEE, 2018.

dataset	type	visual modality	hand-object int	hand-hand it	view	#sub	#obj	#frames	resolution	mesh	labeling	year
Panoptic studio [7]	Real	RGB	v	v	3rd	-	-	15K	1920 x 1080	x	automatic	2017
BigHand2.2M [26]	Real	Depth	x	x	3rd	10	-	2200K	640 x 480	x	magnetic sensor	2017
FPHA [4]	Real	RGB-D	x	v	ego	6	4	105K	1920 x 1080	x	magnetic sensor	2018
GANerated Hands [14]	Synthetic	RGB	v	x	both	-	-	330K	256 x 256	x	synthetic	2018
Rendered Hand Pose [29]	Synthetic	RGB-D	x	x	3rd	20	-	44K	320 x 320	x	synthetic	2017
SynthHands [15]	Synthetic	RGB-D	v	x	ego	-	7	220K	640 x 480	x	synthetic	2017
HO-3D [5]	Real	RGB-D	v	x	3rd	10	10	77K	640 x 480	v	automatic	2021
DexYCB [3]	Real	RGB-D	v	x	3rd	10	20	582K	640 x 480	v	manual	2021
FreiHAND [30]	Real	RGB	v	x	3rd	32	27	37K	224 x 224	v	semi-auto	2019
InterHand2.6M [13]	Real	RGB	x	v	3rd	27	-	2590K	512 x 334	v	semi-auto	2020
Youtube 3D Hands [8]	Real	RGB	v	x	3rd	-	-	50K	-	v	automatic	2020
ContactPose [1]	Real	RGB-D	v	x	3rd	50	25	2991K	960 x 540	v	mocap+thermal	2020
ObMan [6]	Synthetic	RGB-D	v	x	-	20	-	150K	256 x 256	v	synthetic	2019

Table 1. THE COMPARISON BETWEEN DIFFERENT DATASETS

[12] Andrew T Miller and Peter K Allen. Graspit! a versatile simulator for robotic grasping. *IEEE Robotics & Automation Magazine*, 11(4):110–122, 2004.

[13] Gyeongsik Moon, Shoou-I Yu, He Wen, Takaaki Shiratori, and Kyoung Mu Lee. Interhand2. 6m: A dataset and baseline for 3d interacting hand pose estimation from a single rgb image. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, pages 548–564. Springer, 2020.

[14] Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. Ganerated hands for real-time 3d hand tracking from monocular rgb. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 49–59, 2018.

[15] Franziska Mueller, Dushyant Mehta, Oleksandr Sotnychenko, Srinath Sridhar, Dan Casas, and Christian Theobalt. Real-time hand tracking under occlusion from an egocentric rgb-d sensor. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1154–1163, 2017.

[16] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics (ToG)*, 36(6):1–17, 2017.

[17] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[18] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[19] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1145–1153, 2017.

[20] Fisher Yu Yinda Zhang Shuran Song and Ari Seff Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

[21] Srinath Sridhar, Franziska Mueller, Antti Oulasvirta, and Christian Theobalt. Fast and robust hand tracking using detection-guided optimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3213–3221, 2015.

[22] Omid Taheri, Nima Ghorbani, Michael J Black, and Dimitrios Tzionas. Grab: A dataset of whole-body human grasping of objects. In *European Conference on Computer Vision*, pages 581–600. Springer, 2020.

[23] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019.

[24] Gul Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 109–117, 2017.

[25] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017.

[26] Shanxin Yuan, Qi Ye, Bjorn Stenger, Siddhant Jain, and Tae-Kyun Kim. Bighand2. 2m benchmark: Hand pose dataset and state of the art analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4866–4874, 2017.

[27] Yinqiang Zheng, Shigeki Sugimoto, and Masatoshi Okutomi. Aspnp: An accurate and scalable solution to the perspective-n-point problem. *IEICE TRANSACTIONS on Information and Systems*, 96(7):1525–1535, 2013.

[28] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

[29] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single rgb images. In *Proceedings of the IEEE international conference on computer vision*, pages 4903–4911, 2017.

[30] Christian Zimmermann, Duygu Ceylan, Jimei Yang, Bryan Russell, Max Argus, and Thomas Brox. Freihand: A dataset for markerless capture of hand pose and shape from single rgb images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 813–822, 2019.

저자 색인