SHORT-PAPER

# PyG-SSL: A Graph Self-Supervised Learning Toolkit

**LECHENG ZHENG**, University of Illinois Urbana-Champaign, Urbana, IL, United States

**BAOYU JING**, University of Illinois Urbana-Champaign, Urbana, IL, United States

**ZIHAO LI**, University of Illinois Urbana-Champaign, Urbana, IL, United States

**ZHICHEN ZENG**, University of Illinois Urbana-Champaign, Urbana, IL, United States

**TIANXIN WEI**, University of Illinois Urbana-Champaign, Urbana, IL, United States

**MENGTING AI**, University of Illinois Urbana-Champaign, Urbana, IL, United States

View all

**Open Access Support** provided by:

**University of Illinois Urbana-Champaign**

**Wayne State University**

# PyG-SSL: A Graph Self-Supervised Learning Toolkit

**Lecheng Zheng***
lecheng4@illinois.edu
University of Illinois
Urbana-Champaign
Champaign, USA

**Baoyu Jing***
baoyuj2@illinois.edu
University of Illinois
Urbana-Champaign
Champaign, USA

**Zihao Li***
zihaoli5@illinois.edu
University of Illinois
Urbana-Champaign
Champaign, USA

**Zhichen Zeng**
zhichenz@illinois.edu
University of Illinois
Urbana-Champaign
Champaign, USA

**Tianxin Wei**
twei10@illinois.edu
University of Illinois
Urbana-Champaign
Champaign, USA

**Mengting Ai**
mai10@illinois.edu
University of Illinois
Urbana-Champaign
Champaign, USA

**Xinrui He**
xhe33@illinois.edu
University of Illinois
Urbana-Champaign
Champaign, USA

**Lihui Liu**
hw6926@wayne.edu
Wayne State University
Detroit, USA

**Dongqi Fu**
dongqifu@meta.com
Meta AI
Menlo Park, USA

**Jiaxuan You**
jiaxuan@illinois.edu
University of Illinois
Urbana-Champaign
Champaign, USA

**Hanghang Tong**
htong@illinois.edu
University of Illinois
Urbana-Champaign
Champaign, USA

**Jingrui He**
jingrui@illinois.edu
University of Illinois
Urbana-Champaign
Champaign, USA

## Abstract

Graph Self-Supervised Learning (SSL) has emerged as a pivotal area of research in recent years. By engaging in pretext tasks to learn the intricate topological structures and properties of graphs using unlabeled data, these graph SSL models achieve enhanced performance, improved generalization, and heightened robustness. Despite the remarkable achievements of these graph SSL methods, their current implementation poses significant challenges for beginners and practitioners due to the complex nature of graph structures, inconsistent evaluation metrics, and concerns regarding reproducibility hinder further progress in this field. Recognizing the growing interest within the research community, there is an urgent need for a comprehensive, beginner-friendly, and accessible toolkit consisting of the most representative graph SSL algorithms. To address these challenges, we present a Graph SSL toolkit named PyG-SSL, which is built upon PyTorch and is compatible with various deep learning and scientific computing backends. Within the toolkit, we offer a unified framework encompassing dataset loading, hyper-parameter configuration, model training, and comprehensive performance evaluation for diverse downstream tasks. Moreover, we provide beginner-friendly tutorials and the best hyper-parameters of each graph SSL algorithm on different graph datasets, facilitating the reproduction of results. The GitHub repository of the library is https://github.com/iDEA-iSAIL-Lab-UIUC/pyg-ssl.

## CCS Concepts

• **Computing methodologies → Unsupervised learning**; • **Information systems → Data mining**.

## Keywords

Graph Neural Network, Self-supervised Learning, Contrastive Learning

## 1 Introduction and Motivation

Graph Self-Supervised Learning (SSL) has emerged as a pivotal area of research in recent years. By engaging in pretext tasks to learn the intricate topological structures and properties of graphs using unlabeled data [49], these graph SSL models achieve enhanced performance, improved generalization, and heightened robustness [3, 9, 12, 19, 58, 66, 83, 90, 97, 100]. Leveraging the representations learned through SSL, graph downstream tasks like node classification, similarity search, and graph classification benefit significantly [5, 6, 15, 16, 21, 33, 57, 60, 63, 67, 68, 84]. For instance,

---

*Equal Contribution.

Deep Graph InfoMax (DGI), proposed by Velickovic et al. [65] aim to maximize the global-local mutual information and capture the high-level summaries of the graph. Mo et al. [53] propose a multi-loss approach to capitalize on structural and neighbor information, thus enriching interclass variation. GraphCL [84] designs various graph augmentation strategies to augment the raw graph and then maximize the mutual information between two augmented graphs via InfoNCE-style contrastive loss. Hou et al. [18] introduce a masked graph autoencoder with generative self-supervised graph learning, specifically tailored to address issues inherent in graph autoencoders.

Despite the remarkable achievements of these graph SSL methods, their current implementations pose significant challenges for beginners and practitioners due to the complex nature of graph structures and inconsistent evaluation metrics. Additionally, concerns regarding reproducibility continue to hinder progress in this field. Recognizing the growing interest within the research community, there is an urgent need for a comprehensive, beginner-friendly, accessible toolkit that includes the most representative graph SSL algorithms.

To address these challenges, we introduce a Graph SSL toolkit, which includes a diverse set of state-of-the-art graph self-supervised learning methods tailored for various downstream tasks, such as node classification and graph classification. The toolkit offers a unified framework for dataset loading, hyper-parameter configuration, model training, and comprehensive performance evaluation. Built on PyTorch, it is compatible with various deep learning and scientific computing backends, including Numpy, PyTorch Geometric, and DGL. We summarize the main contributions as follows. (1) We present the PyTorch Graph Self-Supervised Learning Toolkit (PyG-SSL), an open-source library for graph self-supervised learning, complete with public releases, documentation, code examples, and continuous integration. (2) We provide a unified framework within PyG-SSL for dataset loading, hyper-parameter configuration, model training, and performance evaluation across various downstream tasks. (3) We offer beginner-friendly tutorials and optimal hyper-parameters for different graph SSL algorithms on various datasets to facilitate result reproduction. (4) We evaluate the performance of numerous graph self-supervised learning methods, providing insights for selecting appropriate methods for specific tasks.

## 2 Comparison to Related Libraries

Table 1 compares the key features of existing graph SSL toolkits, where most of them are open-source code delivered with technical papers. DIG-SSL [103] is a unified and highly customizable framework for implementing graph SSL methods. PyGCL [75] is a graph contrastive learning library. These two packages bear the following limitations: (i) They only support a few SSL algorithms, i.e., 4 in DIG-SSL and 7 in PyGCL, and all these SSL algorithms are designed only for homogeneous graphs. Different from them, our toolkit supports many more SSL algorithms for various types of graphs, including homogeneous graphs, heterogeneous graphs, and molecular graphs; (ii) PyGCL does not provide user-friendly tutorials, which poses great challenges to beginners; (iii) Graph augmentation is a key feature in graph self-supervised learning [83], which is missing in DIG-SSL; (iv) Our toolkit addresses the crucial reproducibility by offering the best hyper-parameters for different
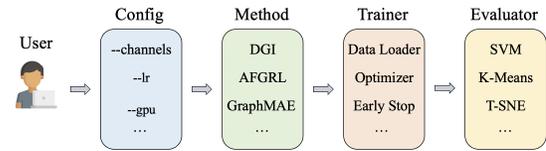


**Figure 1: Overview of PyG-SSL.**

datasets in configuration files, which is absent in both DIG-SSL and PyGCL; (v) Finally, we provide comprehensive documentation[1] to facilitate ease of use and adoption.

## 3 Related Works

Graphs are ubiquitous and are widely used for ranking [1, 8, 13, 34, 35, 56] , social network analysis [7, 10, 11, 14, 76, 78, 79, 86–88, 91–93, 99, 101, 102] , recommendation [2, 17, 26, 37, 48, 62, 70–73, 82, 89, 105] , question answering [38, 40–47, 50], spatial-temporal modeling [24, 29, 30, 39, 64, 81] and text summarization [28, 55] etc. Graph Representation Learning (GRL) aims to transform graph-structured data into low-dimensional representations, while preserving the graph's structural and semantic information. Contrastive learning [25, 69, 95, 98] has garnered significant attention from researchers for its exceptional performance in modeling unlabeled data. Many recent works on graph contrastive learning [22, 23, 27, 36, 74, 80, 83, 84, 96] demonstrate the effectiveness of applying contrastive learning to learn the high quality representation in unsupervised, semi-supervised and transfer learning settings. For instance, GraphCL [84], JOAO [83] and GCA [104] study the impact of graph augmentations when minimizing graph contrastive learning loss. Graph Contrastive Coding (GCC) [58] uses contrastive learning for subgraph instance discrimination within and across networks to capture universal topological properties across multiple networks. MVGRL [12] demonstrates that learning representation by contrasting node and graph across different views outperforms approaches that contrast graph-graph or multi-scale encodings. InfoGraph [60] maximizes the mutual information between the graph-level representation and the representations of substructures of different scales. SUGRL [53] enhances inter-class variation by integrating structural and neighbor information while minimizing intra-class variation using an upper bound loss, achieving efficiency by eliminating data augmentation and discriminators in contrastive learning. This paper aims to offer a unified framework for dataset loading, hyper-parameter configuration, model training, and comprehensive performance evaluation of these methods.

## 4 Design of PyG-SSL

Our PyG-SSL is designed as a user-friendly Python library for general audiences of beginners and practitioners who are interested in graph SSL methods and the applications of these methods in downstream tasks. As illustrated in Figure 1, our framework can be divided into four components:

- **Configuration**: Users can specify detailed configurations for GNNs, SSL methods, etc.
- **Method**: A variety of SSL methods, e.g., DGI, can be called to pre-train GNNs.

---

[1] https://pygssl-tutorial.readthedocs.io/en/latest/index.html

**Table 1: Comparison among different graph SSL toolkits.**

| | # of SSL Algorithms | Augmentation Module | Various Evaluation Metrics | Various Graph Types | Beginner-friendly Tutorials | Formal Documentation |
|---|---|---|---|---|---|---|
| Ours | 10 | ✓ | ✓ | ✓ | ✓ | ✓ |
| DIG-SSL | 4 | ✗ | ✓ | ✗ | ✓ | ✓ |
| PyGCL | 7 | ✓ | ✗ | ✗ | ✗ | ✗ |

- **Trainer**: A trainer that automatically trains the graph SSL method based on the provided configurations.
- **Evaluator**: Multiple downstream evaluators, such as support vector machine (SVM), are readily available for users to assess the performance of pre-trained GNNs.

In addition, our library provides various augmentations, similarity functions, and losses used in the SSL literature for users to build their own SSL methods. Under the MIT License, our toolkit relies solely on open-source libraries. We provide a detailed explanation of each component in the framework.

**Configuration.** PyG-SSL provides an easy-to-use configuration function to load the hyper-parameters for data loader, model, optimizer, and classifier. The configurations of the data loader include the dataset name, root path to load the dataset, and batch size. The configurations of the optimizer contain the learning rate, optimizer name, maximal training epoch, patience epoch, and weight decay. Model configurations mainly consist of default hyperparameters for training. The configurations of the classifier include the hyper-parameters for model evaluation in the downstream task. To facilitate the reproduction of results, we provide beginner-friendly tutorials and the optimal hyper-parameters of each graph SSL algorithm on different graph datasets.

**Method.** The method module in PyG-SSL aims to integrate the major functions of different graph self-supervised techniques. Specifically, the graph self-supervised techniques include augmentation methods for graph contrastive learning (e.g., data augmentation [83, 84] or embedding augmentation [85]), graph encoders to learn hidden representation (e.g., GIN [77], GCN [31], GAT [65]) and various loss functions for model training (e.g., InfoNCE-style loss [84], masked reconstruction loss [18], etc.). The implemented graph self-supervised methods include DGI [66], GraphCL [84], AFGRL [32], MVGRL [12], GCA [104], BGRL [61], SUGRL [53], ReGCL [20], InfoGraph [60], GraphMAE [18], etc.

**Trainer.** The trainer module takes the data loader and the graph self-supervised learning as input and trains the model until convergence. Additionally, we implement early stopping criteria to reduce the computational cost by allowing a limited number of additional epochs if training loss no longer decreases.

**Evaluator.** PyG-SSL integrates task-specific evaluation methods, evaluation metrics and utilities for various purposes. Specifically, we provide traditional classifiers for supervised evaluation, including logistic regression, SVM, random forests, etc; the unsupervised evaluation includes k-mean clustering and similarity search. Our library also provides some utility functions tailored to the research papers we have implemented, such as embedding visualization, allowing users to easily visualize embeddings learned by different graph self-supervised methods.

```python
# --------------- Configuration -------------
torch.manual_seed(0)
config = yaml.safe_load(open("./configuration/merit.yml", 'r',
                         encoding='utf-8').read())
dataset = Amazon(root="pyg_data", name="photo",
                 pre_transform=Compose([NormalizeFeatures(ord=1),
                 Edge2Adj(norm=GCNNorm(add_self_loops=1))]))
data_loader = DataLoader(dataset)
data = dataset.data
# ----------------- Method -----------------
encoder = GCN(in_ft=data.x.shape[1], out_ft=config.model.out_channels,
              projection_hidden_size=config.model.projection_hidden_size,
              projection_size=config.model.projection_size)
method = Merit(encoder=encoder, data=data, config=config,
               device="cuda", is_sparse=True)
# ----------------- Trainer -----------------
trainer = SimpleTrainer(method=method, data_loader=data_loader,
                        lr=config.model.lr, weight_decay=config.model.weight_decay,
                        max_iter=config.model.num_epoch, device="cuda")
trainer.train()
# ----------------- Evaluator -----------------
data_pyg = dataset.data.to(method.device)
embs = method.get_embs(data_pyg, data_pyg.adj_t).detach()
lg = LogisticRegression(lr=config.optim.lr, weight_decay=config.optim.weight_decay,
                        max_iter=config.optim.num_epoch, n_run=50, device="cuda")
create_masks(data=data_pyg.cpu())
lg(embs=embs, dataset=data_pyg)
```

**Figure 2: Example of Implementation.**

## 4.1 Documentation and Code Example

We release our software with publicly available documentation at https://pygssl-tutorial.readthedocs.io/en/latest/. Our documentation covers the four major components of PyG-SSL and graph self-supervised methods in various research papers. The documentation also includes an in-depth tutorial guide and a tour of external resources. In our GitHub repository, we offer code examples for all papers whose methods have been implemented in our library. The documentation also features examples of configuration, method setup, model training, and evaluation. One such example is visualized in Figure 2.

**Table 2: Dataset Statistics**

| Dataset | # of graphs | # of nodes | # of edges | # of features | # of classes |
|---|---|---|---|---|---|
| WikiCS | 1 | 11,701 | 216,123 | 0 | 10 |
| Coauthor | 1 | 18,333 | 163,788 | 6,805 | 15 |
| Amazon | 1 | 7,650 | 238,162 | 745 | 8 |
| IMDB-B | 1,000 | 19.8 | 193.1 | 0 | 2 |
| IMDB-M | 1,500 | 13 | 26.4 | 0 | 3 |
| Mutag | 188 | 17.9 | 39.6 | 7 | 2 |

## 5 Experimental Evaluation

**Datasets.** PyG-SSL evaluates the performance of graph self-supervised methods on six widely-used datasets, including WikiCS [52], Coauthor [59], Amazon-Photo [51], IMDB-B [4], IMDB-M [94] and Mutag [54]. The data statistics of these datasets are summarized in Table 2. To demonstrate that the implemented methods can indeed reproduce the performance in the original papers, we evaluate

**Table 3: Reproducibility of graph SSL algorithms on the node classification task. '-' means that the result is not reported in the research paper.**

| Methods | Implementation | DGI | GraphCL | AFGRL | MVGRL | GCA | BGRL | SUGRL | ReGCL |
|---|---|---|---|---|---|---|---|---|---|
| WikiCS | Reported | - | - | 77.6±0.5 | - | 78.3±0.0 | 80.0±0.1 | - | - |
| | Ours | 75.3±1.1 | 77.3±0.8 | 78.9±0.6 | 72.7±0.9 | 77.6±0.5 | 78.3±0.7 | 70.0±1.1 | 78.7±0.7 |
| Coauthor | Reported | - | - | 93.3±0.2 | - | 93.1±0.0 | 93.3±0.1 | - | 93.7±0.3 |
| | Ours | 93.5±0.3 | 84.0±0.6 | 92.1±0.2 | 85.5±0.4 | 91.6±0.2 | 92.4±0.2 | 92.9±0.1 | 91.2±0.2 |
| Amazon | Reported | - | - | 93.2±0.3 | - | 92.5±0.1 | 93.2±0.3 | 93.2±0.2 | 92.6±0.3 |
| | Ours | 92.3±0.7 | 81.7±0.9 | 92.6±0.3 | 86.5±1.2 | 91.2±0.3 | 92.6±0.3 | 91.7±0.2 | 91.6±0.4 |

**Table 4: Reproducibility of graph SSL algorithms on the graph classification task. '-' denotes that the result is not reported in the research paper.**

| Methods | Implementation | InfoGraph | GraphCL | GraphMAE |
|---|---|---|---|---|
| IMDB-B | Reported | 73.0±0.9 | 71.1±0.4 | - |
| | Ours | 72.2±2.1 | 71.5±3.6 | 72.6±3.1 |
| IMDB-M | Reported | 49.7±0.5 | - | - |
| | Ours | 49.5±2.7 | 48.0±3.8 | 49.6±2.5 |
| Mutag | Reported | 89.0±1.1 | 86.8±1.3 | - |
| | Ours | 88.8±5.0 | 86.7±4.6 | 86.2±4.1 |

the task performance of all graph self-supervised learning methods implemented in the library. We report the mean and standard deviation for each result, where ±0.0 indicates that the standard deviation is less than 0.05%. Our implementations generally reproduce the results in the original papers, but slight differences (negligible considering standard deviations) may occur due to data splits or random seeds. In addition, we provide insights into which methods are preferred for node classification and graph classification tasks.

**Node Classification.** Table 3 presents the performance of various graph self-supervised learning methods on the node classification task across three datasets. The following observations can be made: (1) Some methods excel on specific datasets but do not perform consistently well across all datasets. Specifically, SUGRL achieves the highest performance on the Coauthor dataset but performs poorly on the WikiCS dataset, whereas MVGRL shows the best performance on the WikiCS dataset. (2) AFGRL, BGRL, and DGI rank among the top three performers on average across the three datasets. Unlike many contrastive learning-based methods (e.g., GraphCL, GCA), AFGRL and BGRL are non-contrastive methods that do not require negative samples during training. AFGRL utilizes simple augmentations, such as node feature and edge masking, while BGRL is augmentation-free, generating an alternative view of a graph by identifying nodes that share local structural information and global semantics. Similarly, DGI explores local-global consistency by maximizing local-global mutual information.

**Graph Classification.** Table 4 shows the performance of numerous graph self-supervised learning methods on the graph classification task. Notice that though GraphMAE reports the accuracy on these three datasets, GraphMAE only calculates the F1 score in their source code implementation. Thus, we do not include the reported score in Table 4. By observation, we find that GraphMAE outperforms InfoGraph and GraphCL on the IMDB-B and IMDB-M

datasets. We attribute this superior performance to the design of predicting masked features, which addresses some of the limitations inherent in generative self-supervised graph pretraining methods.

**Suggestions for Beginner.** Among these self-supervised graph pre-training methods, GraphCL incorporates a variety of graph augmentation strategies and is optimized using a standard contrastive learning objective. Its flexibility makes it suitable for a range of tasks, including node classification, link prediction, and graph classification. For node classification or link prediction tasks, BGRL is a strong starting point due to its competitive performance and scalability to large-scale graphs. For graph-level classification tasks, we recommend beginners start with InfoGraph, which serves as a base model for many subsequent methods, including GraphCL and MR-GRL.

## 6 Conclusion

In this paper, we present the Graph Self-Supervised Learning Toolkit (PyG-SSL), a user-friendly library designed to streamline and advance research in graph self-supervised learning. Our toolkit offers a comprehensive solution for implementing and experimenting with a wide array of state-of-the-art graph SSL methods. With robust support for various graph types and a unified framework that simplifies tasks such as dataset loading, hyperparameter tuning, model training, and performance evaluation, PyG-SSL empowers both beginners and experienced practitioners to efficiently explore and apply graph SSL techniques. The evaluations of numerous graph SSL methods included in PyG-SSL offer practical insights for selecting the most suitable approaches for different tasks. PyG-SSL not only stands as a significant advancement in graph SSL but also complements other related libraries, fostering a collaborative and evolving landscape in graph-based research and applications.

## Acknowledgments

## 7 GenAI Usage Disclosure

In this paper, generative AI tool is used to edit and improve the quality of the text, including checking the spelling, grammar, punctuation and clarity.

## References

[1] Yikun Ban, Yunzhe Qi, Tianxin Wei, Lihui Liu, and Jingrui He. 2024. Meta Clustering of Neural Bandits. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024*. ACM, 95–106.

[2] Yikun Ban, Jiaru Zou, Zihao Li, Yunzhe Qi, Dongqi Fu, Jian Kang, Hanghang Tong, and Jingrui He. 2024. PageRank Bandits for Link Prediction. *CoRR* abs/2411.01410 (2024). arXiv:2411.01410 doi:10.48550/ARXIV.2411.01410

[3] Wenxuan Bao, Zhichen Zeng, Zhining Liu, Hanghang Tong, and Jingrui He. 2024. Matcha: Mitigating Graph Structure Shifts with Test-Time Adaptation. *arXiv preprint arXiv:2410.06976* (2024).

[4] Chen Cai and Yusu Wang. 2018. A simple yet effective baseline for non-attributed graph classification. *arXiv preprint arXiv:1811.03508* (2018).

[5] Shengyu Feng, Baoyu Jing, Yada Zhu, and Hanghang Tong. 2022. Adversarial graph contrastive learning with information regularization. In *Proceedings of the ACM Web Conference 2022*. 1362–1371.

[6] Shengyu Feng, Baoyu Jing, Yada Zhu, and Hanghang Tong. 2024. Ariel: Adversarial graph contrastive learning. *ACM Transactions on Knowledge Discovery from Data* 18, 4 (2024), 1–22.

[7] Dongqi Fu, Liri Fang, Zihao Li, Hanghang Tong, Vetle I. Torvik, and Jingrui He. 2024. Parametric Graph Representations in the Era of Foundation Models: A Survey and Position. *CoRR* abs/2410.12126 (2024). arXiv:2410.12126 doi:10.48550/ARXIV.2410.12126

[8] Dongqi Fu, Zhigang Hua, Yan Xie, Jin Fang, Si Zhang, Kaan Sancak, Hao Wu, Andrey Malevich, Jingrui He, and Bo Long. 2024. VCR-Graphormer: A Mini-batch Graph Transformer via Virtual Connections. In *ICLR*.

[9] Dongqi Fu, Zhe Xu, Bo Li, Hanghang Tong, and Jingrui He. 2020. A View-Adversarial Framework for Multi-View Network Embedding. In *CIKM*.

[10] Dongqi Fu, Dawei Zhou, and Jingrui He. 2020. Local Motif Clustering on Time-Evolving Graphs. In *KDD*.

[11] Dongqi Fu, Dawei Zhou, Ross Maciejewski, Arie Croitoru, Marcus Boyd, and Jingrui He. 2023. Fairness-Aware Clique-Preserving Spectral Clustering of Temporal Graphs. In *WWW*.

[12] Kaveh Hassani and Amir Hosein Khas Ahmadi. 2020. Contrastive Multi-View Representation Learning on Graphs. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event (Proceedings of Machine Learning Research, Vol. 119)*. PMLR, 4116–4126.

[13] Jingrui He, Hanghang Tong, Qiaozhu Mei, and Boleslaw K. Szymanski. 2012. GenDeR: A Generic Diversified Ranking Algorithm. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Léon Bottou, and Kilian Q. Weinberger (Eds.). 1151–1159. https://proceedings.neurips.cc/paper/2012/hash/7f24d240521d99071c93af3917215ef7-Abstract.html

[14] Xinrui He, Yikun Ban, Jiaru Zou, Tianxin Wei, Curtiss Cook, and Jingrui He. 2025. LLM-Forest: Ensemble Learning of LLMs with Graph-Augmented Prompts for Data Imputation. In *Findings of the Association for Computational Linguistics: ACL 2025*, Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (Eds.). Association for Computational Linguistics, Vienna, Austria, 6921–6936. doi:10.18653/v1/2025.findings-acl.361

[15] Xinyu He, Dongqi Fu, Hanghang Tong, Ross Maciejewski, and Jingrui He. 2025. Temporal Heterogeneous Graph Generation with Privacy, Utility, and Efficiency. In *ICLR*.

[16] Xinrui He, Shuo Liu, Jacky Keung, and Jingrui He. 2024. Co-clustering for Federated Recommender System. In *Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, May 13-17, 2024*, Tat-Seng Chua, Chong-Wah Ngo, Ravi Kumar, Hady W. Lauw, and Roy Ka-Wei Lee (Eds.). ACM, 3821–3832. doi:10.1145/3589334.3645626

[17] Xinrui He, Tianxin Wei, and Jingrui He. 2023. Robust Basket Recommendation via Noise-tolerated Graph Contrastive Learning. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management* (Birmingham, United Kingdom) *(CIKM '23)*. Association for Computing Machinery, New York, NY, USA, 709–719. doi:10.1145/3583780.3615039

[18] Zhenyu Hou, Xiao Liu, Yukuo Cen, Yuxiao Dong, Hongxia Yang, Chunjie Wang, and Jie Tang. 2022. GraphMAE: Self-Supervised Masked Graph Autoencoders. In *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, Aidong Zhang and Huzefa Rangwala (Eds.). ACM, 594–604.

[19] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay S. Pande, and Jure Leskovec. 2020. Strategies for Pre-training Graph Neural Networks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

[20] Cheng Ji, Zixuan Huang, Qingyun Sun, Hao Peng, Xingcheng Fu, Qian Li, and Jianxin Li. 2024. ReGCL: Rethinking Message Passing in Graph Contrastive Learning. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan (Eds.). AAAI Press, 8544–8552.

[21] Ming Jin, Yizhen Zheng, Yuan-Fang Li, Chen Gong, Chuan Zhou, and Shirui Pan. 2021. Multi-Scale Contrastive Siamese Networks for Self-Supervised Graph Representation Learning. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, Zhi-Hua Zhou (Ed.). ijcai.org, 1477–1483.

[22] Baoyu Jing, Shengyu Feng, Yuejia Xiang, Xi Chen, Yu Chen, and Hanghang Tong. 2022. X-GOAL: Multiplex heterogeneous graph prototypical contrastive learning. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 894–904.

[23] Baoyu Jing, Chanyoung Park, and Hanghang Tong. 2021. Hdmi: High-order deep multiplex infomax. In *Proceedings of the Web Conference 2021*. 2414–2424.

[24] Baoyu Jing, Hanghang Tong, and Yada Zhu. 2021. Network of tensor time series. In *Proceedings of the Web Conference 2021*. 2425–2437.

[25] Baoyu Jing, Yansen Wang, Guoxin Sui, Jing Hong, Jingrui He, Yuqing Yang, Dongsheng Li, and Kan Ren. 2024. Automated Contrastive Learning Strategy Search for Time Series. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 4612–4620.

[26] Baoyu Jing, Yuchen Yan, Kaize Ding, Chanyoung Park, Yada Zhu, Huan Liu, and Hanghang Tong. 2024. Sterling: Synergistic representation learning on bipartite graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 12976–12984.

[27] Baoyu Jing, Yuchen Yan, Yada Zhu, and Hanghang Tong. [n. d.]. Coin: Co-cluster infomax for bipartite graphs. In *NeurIPS 2022 Workshop: New Frontiers in Graph Learning*.

[28] Baoyu Jing, Zeyu You, Tao Yang, Wei Fan, and Hanghang Tong. 2021. Multiplex Graph Neural Network for Extractive Text Summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 133–139.

[29] Baoyu Jing, Si Zhang, Yada Zhu, Bin Peng, Kaiyu Guan, Andrew Margenot, and Hanghang Tong. 2022. Retrieval based time series forecasting. *arXiv preprint arXiv:2209.13525* (2022).

[30] Baoyu Jing, Dawei Zhou, Kan Ren, and Carl Yang. 2024. Causality-Aware Spatiotemporal Graph Neural Networks for Spatiotemporal Time Series Imputation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 1027–1037.

[31] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

[32] Namkyeong Lee, Junseok Lee, and Chanyoung Park. 2022. Augmentation-free self-supervised learning on graphs. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 36. 7372–7380.

[33] Bolian Li, Baoyu Jing, and Hanghang Tong. 2022. Graph communal contrastive learning. In *Proceedings of the ACM web conference 2022*. 1203–1213.

[34] Zihao Li, Dongqi Fu, Mengting Ai, and Jingrui He. 2024. APEX$^2$: Adaptive and Extreme Summarization for Personalized Knowledge Graphs. *arXiv preprint arXiv:2412.17336* (2024).

[35] Zihao Li, Dongqi Fu, and Jingrui He. 2023. Everything Evolves in Personalized PageRank. In *Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023*, Ying Ding, Jie Tang, Juan F. Sequeda, Lora Aroyo, Carlos Castillo, and Geert-Jan Houben (Eds.). ACM, 3342–3352. doi:10.1145/3543507.3583474

[36] Zihao Li, Lecheng Zheng, Bowen Jin, Dongqi Fu, Baoyu Jing, Yikun Ban, Jingrui He, and Jiawei Han. 2024. Can Graph Neural Networks Learn Language with Extremely Weak Text Supervision? *arXiv preprint arXiv:2412.08174* (2024).

[37] Mingfu Liang, Xi Liu, Rong Jin, Boyang Liu, Qiuling Suo, Qinghai Zhou, Song Zhou, Laming Chen, Hua Zheng, Zhiyuan Li, et al. 2025. External large foundation model: How to efficiently serve trillions of parameters for online ads recommendation. In *Companion Proceedings of the ACM on Web Conference 2025*. 344–353.

[38] Jiacheng Lin, Kun Qian, Haoyu Han, Narendra Choudhary, Tianxin Wei, Zhongruo Wang, Sahika Genc, Edward W Huang, Sheng Wang, Karthik Subbian, Danai Koutra, and Jimeng Sun. 2024. GT2Vec: Large Language Models as Multi-Modal Encoders for Text and Graph-Structured Data. *arXiv preprint arXiv:2410.11235* (2024). https://arxiv.org/abs/2410.11235

[39] Xiao Lin, Zhichen Zeng, Tianxin Wei, Zhining Liu, Hanghang Tong, et al. 2025. Cats: Mitigating correlation shift for multivariate time series classification. *arXiv preprint arXiv:2504.04283* (2025).

[40] Lihui Liu, Yuzhong Chen, Mahashweta Das, Hao Yang, and Hanghang Tong. 2023. Knowledge Graph Question Answering with Ambiguous Query. In *Proceedings of the ACM Web Conference 2023*.

[41] Lihui Liu, Boxin Du, Yi Ren Fung, Heng Ji, Jiejun Xu, and Hanghang Tong. 2021. Kompare: A knowledge graph comparative reasoning system. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 3308–3318.

[42] Lihui Liu, Boxin Du, Heng Ji, ChengXiang Zhai, and Hanghang Tong. 2021. Neural-answering logical queries on knowledge graphs. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*. 1087–1097.

[43] Lihui Liu, Boxin Du, Jiejun Xu, Yinglong Xia, and Hanghang Tong. 2022. Joint Knowledge Graph Completion and Question Answering. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Washington DC, USA) *(KDD '22)*. Association for Computing Machinery, New York, NY, USA, 1098–1108.

[44] Lihui Liu, Blaine Hill, Boxin Du, Fei Wang, and Hanghang Tong. 2024. Conversational Question Answering with Language Models Generated Reformulations over Knowledge Graph. In *Findings of the Association for Computational Linguistics: ACL 2024*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 839–850. doi:10.18653/v1/2024.findings-acl.48

[45] Lihui Liu and Hanghang Tong. 2023. Knowledge graph reasoning and its applications. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 5813–5814.

[46] Lihui Liu, Zihao Wang, Ruizhong Qiu, Yikun Ban, Eunice Chan, Yangqiu Song, Jingrui He, and Hanghang Tong. 2024. Logic query of thoughts: Guiding large language models to answer complex logic queries with knowledge graphs. *arXiv preprint arXiv:2404.04264* (2024).

[47] Lihui Liu, Zihao Wang, Dawei Zhou, Ruijie Wang, Yuchen Yan, Bo Xiong, Sihong He, Kai Shu, and Hanghang Tong. 2025. TransNet: Transfer Knowledge for Few-shot Knowledge Graph Completion. *arXiv preprint arXiv:2504.03720* (2025).

[48] Xiaolong Liu, Zhichen Zeng, Xiaoyi Liu, Siyang Yuan, Weinan Song, Mengyue Hang, Yiqun Liu, Chaofei Yang, Donghyun Kim, Wen-Yen Chen, et al. 2024. A collaborative ensemble framework for ctr prediction. *arXiv preprint arXiv:2411.13700* (2024).

[49] Yixin Liu, Ming Jin, Shirui Pan, Chuan Zhou, Yu Zheng, Feng Xia, and Philip S. Yu. 2023. Graph Self-Supervised Learning: A Survey. *IEEE Trans. Knowl. Data Eng.* 35, 6 (2023), 5879–5900.

[50] Zhining Liu, Rana Ali Amjad, Ravinarayana Adkathimar, Tianxin Wei, and Hanghang Tong. 2025. SelfElicit: Your language model secretly knows where is the relevant evidence. *arXiv preprint arXiv:2502.08767* (2025).

[51] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*. 43–52.

[52] Péter Mernyei and Cătălina Cangea. 2020. Wiki-cs: A wikipedia-based benchmark for graph neural networks. *arXiv preprint arXiv:2007.02901* (2020).

[53] Yujie Mo, Liang Peng, Jie Xu, Xiaoshuang Shi, and Xiaofeng Zhu. 2022. Simple Unsupervised Graph Representation Learning. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*. AAAI Press, 7797–7805.

[54] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. 2016. Learning convolutional neural networks for graphs. In *International conference on machine learning*. PMLR, 2014–2023.

[55] Xuying Ning, Dongqi Fu, Tianxin Wei, Wenxuan Xu, and Jingrui He. 2025. Graph4MM: Weaving Multimodal Learning with Structural Information. In *Proceedings of the Forty-second International Conference on Machine Learning*.

[56] Lawrence Page. 1999. *The PageRank citation ranking: Bringing order to the web*. Technical Report. Technical Report.

[57] Chanyoung Park, Donghyun Kim, Jiawei Han, and Hwanjo Yu. 2020. Unsupervised Attributed Multiplex Network Embedding. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*. AAAI Press, 5371–5378.

[58] Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. 2020. GCC: Graph Contrastive Coding for Graph Neural Network Pre-Training. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*. ACM, 1150–1160.

[59] Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-June Hsu, and Kuansan Wang. 2015. An overview of microsoft academic service (mas) and applications. In *Proceedings of the 24th international conference on world wide web*. 243–246.

[60] Fan-Yun Sun, Jordan Hoffmann, Vikas Verma, and Jian Tang. 2020. InfoGraph: Unsupervised and Semi-supervised Graph-Level Representation Learning via Mutual Information Maximization. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

[61] Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Mehdi Azabou, Eva L. Dyer, Rémi Munos, Petar Velickovic, and Michal Valko. 2022. Large-Scale Representation Learning on Graphs via Bootstrapping. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

[62] Katherine Tieu, Dongqi Fu, Zihao Li, Ross Maciejewski, and Jingrui He. 2025. Learnable Spatial-Temporal Positional Encoding for Link Prediction. In *ICML*.

[63] Katherine Tieu, Dongqi Fu, Jun Wu, and Jingrui He. 2025. Invariant Link Selector for Spatial-Temporal Out-of-Distribution Problem. In *AISTATS*.

[64] Katherine Tieu, Dongqi Fu, Yada Zhu, Hendrik F. Hamann, and Jingrui He. 2024. Temporal Graph Neural Tangent Kernel with Graphon-Guaranteed. In *NeurIPS*.

[65] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

[66] Petar Velickovic, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R. Devon Hjelm. 2018. Deep Graph Infomax. *CoRR* abs/1809.10341 (2018).

[67] Haohui Wang, Baoyu Jing, Kaize Ding, Yada Zhu, Wei Cheng, Si Zhang, Yonghui Fan, Liqing Zhang, and Dawei Zhou. 2024. Mastering long-tail complexity on graphs: Characterization, learning, and generalization. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3045–3056.

[68] Limei Wang, Kaveh Hassani, Si Zhang, Dongqi Fu, Baichuan Yuan, Weilin Cong, Zhigang Hua, Hao Wu, Ning Yao, and Bo Long. 2025. Learning Graph Quantized Tokenizers. In *ICLR*.

[69] Tianxin Wei, Yifan Chen, Xinrui He, Wenxuan Bao, and Jingrui He. [n. d.]. Connecting Domains and Contrasting Samples: A Ladder for Domain Generalization. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining, V.1, KDD 2025, Toronto, ON, Canada, August 3-7, 2025*.

[70] Tianxin Wei, Fuli Feng, Jiawei Chen, Ziwei Wu, Jinfeng Yi, and Xiangnan He. 2021. Model-agnostic counterfactual reasoning for eliminating popularity bias in recommender system. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*. 1791–1800.

[71] Tianxin Wei and Jingrui He. 2022. Comprehensive Fair Meta-learned Recommender System. In *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, Aidong Zhang and Huzefa Rangwala (Eds.). ACM, 1989–1999. doi:10.1145/3534678.3539269

[72] Tianxin Wei, Bowen Jin, Ruirui Li, Hansi Zeng, Zhengyang Wang, Jianhui Sun, Qingyu Yin, Hanqing Lu, Suhang Wang, Jingrui He, et al. 2024. Towards unified multi-modal personalization: Large vision-language models for generative recommendation and beyond. *arXiv preprint arXiv:2403.10667* (2024).

[73] Tianxin Wei, Ziwei Wu, Ruirui Li, Ziniu Hu, Fuli Feng, Xiangnan He, Yizhou Sun, and Wei Wang. 2020. Fast adaptation for cold-start collaborative filtering with meta-learning. In *2020 IEEE International Conference on Data Mining (ICDM)*. IEEE, 661–670.

[74] Tianxin Wei, Yuning You, Tianlong Chen, Yang Shen, Jingrui He, and Zhangyang Wang. 2022. Augmentations in hypergraph contrastive learning: Fabricated and generative. *Advances in neural information processing systems* 35 (2022), 1909–1922.

[75] Yaochen Xie, Zhao Xu, Jingtun Zhang, Zhengyang Wang, and Shuiwang Ji. 2023. Self-Supervised Learning of Graph Neural Networks: A Unified Review. *IEEE Trans. Pattern Anal. Mach. Intell.* 45, 2 (2023), 2412–2429.

[76] Haobo Xu, Yuchen Yan, Dingsu Wang, Zhe Xu, Zhichen Zeng, Tarek F Abdelzaher, Jiawei Han, and Hanghang Tong. 2024. SLOG: An Inductive Spectral Graph Neural Network Beyond Polynomial Filter. In *Forty-first International Conference on Machine Learning*.

[77] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How Powerful are Graph Neural Networks?. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

[78] Yuchen Yan, Yuzhong Chen, Huiyuan Chen, Xiaoting Li, Zhe Xu, Zhichen Zeng, Lihui Liu, Zhining Liu, and Hanghang Tong. 2024. Thegcn: Temporal heterophilic graph convolutional network. *arXiv preprint arXiv:2412.16435* (2024).

[79] Yuchen Yan, Yongyi Hu, Qinghai Zhou, Lihui Liu, Zhichen Zeng, Yuzhong Chen, Menghai Pan, Huiyuan Chen, Mahashweta Das, and Hanghang Tong. 2024. Pacer: Network embedding from positional to structural. In *Proceedings of the ACM on Web Conference 2024*. 2485–2496.

[80] Yuchen Yan, Baoyu Jing, Lihui Liu, Ruijie Wang, Jinning Li, Tarek Abdelzaher, and Hanghang Tong. 2024. Reconciling competing sampling strategies of network embedding. *Advances in Neural Information Processing Systems* 36 (2024).

[81] Yuchen Yan, Lihui Liu, Yikun Ban, Baoyu Jing, and Hanghang Tong. 2021. Dynamic knowledge graph alignment. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35. 4564–4572.

[82] Hyunsik Yoo, Zhichen Zeng, Jian Kang, Ruizhong Qiu, David Zhou, Zhining Liu, Fei Wang, Charlie Xu, Eunice Chan, and Hanghang Tong. 2024. Ensuring User-side Fairness in Dynamic Recommender Systems. In *Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, May 13-17, 2024*, Tat-Seng Chua, Chong-Wah Ngo, Ravi Kumar, Hady W. Lauw, and Roy Ka-Wei Lee (Eds.). ACM, 3667–3678. doi:10.1145/3589334.3645536

[83] Yuning You, Tianlong Chen, Yang Shen, and Zhangyang Wang. 2021. Graph Contrastive Learning Automated. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, 12121–12132.

[84] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph Contrastive Learning with Augmentations. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual.*

[85] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Lizhen Cui, and Quoc Viet Hung Nguyen. 2022. Are graph augmentations necessary? simple graph contrastive learning for recommendation. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval.* 1294–1303.

[86] Qi Yu, Zhichen Zeng, Yuchen Yan, Zhining Liu, Baoyu Jing, Ruizhong Qiu, Ariful Azad, and Hanghang Tong. 2025. PLANETALIGN: A Comprehensive Python Library for Benchmarking Network Alignment. *arXiv preprint arXiv:2505.21366* (2025).

[87] Qi Yu, Zhichen Zeng, Yuchen Yan, Lei Ying, R Srikant, and Hanghang Tong. 2025. Joint optimal transport and embedding for network alignment. In *Proceedings of the ACM on Web Conference 2025.* 2064–2075.

[88] Zhichen Zeng, Boxin Du, Si Zhang, Yinglong Xia, Zhining Liu, and Hanghang Tong. 2024. Hierarchical multi-marginal optimal transport for network alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 16660–16668.

[89] Zhichen Zeng, Xiaolong Liu, Mengyue Hang, Xiaoyi Liu, Qinghai Zhou, Chaofei Yang, Yiqun Liu, Yichen Ruan, Laming Chen, Yuxin Chen, et al. 2024. Interformer: Towards effective heterogeneous interaction learning for click-through rate prediction. *arXiv preprint arXiv:2411.09852* (2024).

[90] Zhichen Zeng, Ruizhong Qiu, Wenxuan Bao, Tianxin Wei, Xiao Lin, Yuchen Yan, Tarek F Abdelzaher, Jiawei Han, and Hanghang Tong. 2025. Pave Your Own Path: Graph Gradual Domain Adaptation on Fused Gromov-Wasserstein Geodesics. *arXiv preprint arXiv:2505.12709* (2025).

[91] Zhichen Zeng, Ruizhong Qiu, Zhe Xu, Zhining Liu, Yuchen Yan, Tianxin Wei, Lei Ying, Jingrui He, and Hanghang Tong. 2024. Graph mixup on approximate gromov–wasserstein geodesics. In *Forty-first International Conference on Machine Learning.*

[92] Zhichen Zeng, Si Zhang, Yinglong Xia, and Hanghang Tong. 2023. Parrot: Position-aware regularized optimal transport for network alignment. In *Proceedings of the ACM Web Conference 2023.* 372–382.

[93] Zhichen Zeng, Ruike Zhu, Yinglong Xia, Hanqing Zeng, and Hanghang Tong. 2023. Generative graph dictionary learning. In *International Conference on Machine Learning.* PMLR, 40749–40769.

[94] Qi Zhao and Yusu Wang. 2019. Learning metrics for persistence-based summaries and applications for graph classification. *Advances in neural information processing systems* 32 (2019).

[95] Lecheng Zheng, John R. Birge, Haiyue Wu, Yifang Zhang, and Jingrui He. 2025. Cluster Aware Graph Anomaly Detection. In *Proceedings of the ACM on Web Conference 2025, WWW 2025, Sydney, NSW, Australia, 28 April 2025- 2 May 2025*, Guodong Long, Michale Blumestein, Yi Chang, Liane Lewin-Eytan, Zi Helen Huang, and Elad Yom-Tov (Eds.). ACM, 1771–1782.

[96] Lecheng Zheng, Zhengzhang Chen, Jingrui He, and Haifeng Chen. 2024. MU-LAN: Multi-modal Causal Structure Learning and Root Cause Analysis for Microservice Systems. In *Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, May 13-17, 2024*, Tat-Seng Chua, Chong-Wah Ngo, Ravi Kumar, Hady W. Lauw, and Roy Ka-Wei Lee (Eds.). ACM, 4107–4116.

[97] Lecheng Zheng, Dongqi Fu, Ross Maciejewski, and Jingrui He. 2024. DrGNN: Deep Residual Graph Neural Network with Contrastive Learning. In *TMLR.*

[98] Lecheng Zheng, Jinjun Xiong, Yada Zhu, and Jingrui He. 2022. Contrastive Learning with Complex Heterogeneity. In *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, Aidong Zhang and Huzefa Rangwala (Eds.). ACM, 2594–2604.

[99] Lecheng Zheng, Dawei Zhou, Hanghang Tong, Jiejun Xu, Yada Zhu, and Jingrui He. 2024. Fairgen: Towards Fair Graph Generation. In *40th IEEE International Conference on Data Engineering, ICDE 2024, Utrecht, The Netherlands, May 13-16, 2024.* IEEE, 2285–2297.

[100] Dawei Zhou, Lecheng Zheng, Dongqi Fu, Jiawei Han, and Jingrui He. 2022. MentorGNN: Deriving Curriculum for Pre-Training GNNs. In *CIKM.*

[101] Dawei Zhou, Lecheng Zheng, Jiawei Han, and Jingrui He. 2020. A Data-Driven Graph Generative Model for Temporal Interaction Networks. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, Rajesh Gupta, Yan Liu, Jiliang Tang, and B. Aditya Prakash (Eds.). ACM, 401–411.

[102] Dawei Zhou, Lecheng Zheng, Jiejun Xu, and Jingrui He. 2019. Misc-GAN: A Multi-scale Generative Model for Graphs. *Frontiers Big Data* 2 (2019), 3.

[103] Yanqiao Zhu, Yichen Xu, Qiang Liu, and Shu Wu. 2021. An Empirical Study of Graph Contrastive Learning. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, Joaquin Vanschoren and Sai-Kit Yeung (Eds.).

[104] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2021. Graph Contrastive Learning with Adaptive Augmentation. In *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, Jure Leskovec, Marko Grobelnik, Marc Najork, Jie Tang, and Leila Zia (Eds.). ACM / IW3C2, 2069–2080.

[105] Zhuang Zhuang, Tianxin Wei, Lingbo Liu, Heng Qi, Yanming Shen, and Baocai Yin. 2024. TAU: Trajectory Data Augmentation with Uncertainty for Next POI Recommendation. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI.* AAAI Press, 22565–22573.