GSLB: The Graph Structure Learning Benchmark

Zhixun Li[1] **Liang Wang**[2,3] **Xin Sun**[4] **Yifan Luo**[5] **Yanqiao Zhu**[6] **Dingshuo Chen**[2,3]
Yingtao Luo[7] **Xiangxin Zhou**[2,3] **Qiang Liu**[2,3*] **Shu Wu**[2,3] **Liang Wang**[2,3,4] **Jeffrey Xu Yu**[1*]

[1] Department of Systems Engineering and Engineering Management
The Chinese University of Hong Kong
[2]Center for Research on Intelligent Perception and Computing
State Key Laboratory of Multimodal Artificial Intelligence Systems
Institute of Automation, Chinese Academy of Sciences
[3] School of Artificial Intelligence, University of Chinese Academy of Sciences
[4]Department of Automation, University of Science and Technology of China
[5]School of Cyberspace Security, Beijing University of Posts and Telecommunications
[6]Department of Computer Science, University of California, Los Angeles
[7]Heinz College of Information Systems and Public Policy, Machine Learning Department
School of Computer Science, Carnegie Mellon University
✉ Primary contact: `zxli@se.cuhk.edu.hk`

Abstract

Graph Structure Learning (GSL) has recently garnered considerable attention due to its ability to optimize both the parameters of Graph Neural Networks (GNNs) and the computation graph structure simultaneously. Despite the proliferation of GSL methods developed in recent years, there is no standard experimental setting or fair comparison for performance evaluation, which creates a great obstacle to understanding the progress in this field. To fill this gap, we systematically analyze the performance of GSL in different scenarios and develop a comprehensive Graph Structure Learning Benchmark (GSLB) curated from 20 diverse graph datasets and 16 distinct GSL algorithms. Specifically, GSLB systematically investigates the characteristics of GSL in terms of three dimensions: **effectiveness**, **robustness**, and **complexity**. We comprehensively evaluate state-of-the-art GSL algorithms in node- and graph-level tasks, and analyze their performance in robust learning and model complexity. Further, to facilitate reproducible research, we have developed an easy-to-use library for training, evaluating, and visualizing different GSL methods. Empirical results of our extensive experiments demonstrate the ability of GSL and reveal its potential benefits on various downstream tasks, offering insights and opportunities for future research. The code of GSLB is available at: `https://github.com/GSL-Benchmark/GSLB`.

1 Introduction

Graphs, structures made of vertices and edges, are ubiquitous in real-world applications. A wide variety of applications spanning social network [51, 9], molecular property prediction [40, 14], fake news detection [45, 1], and fraud detection [23, 27] have found graphs instrumental in modeling complex systems. In recent years, Graph Neural Networks (GNNs) have attracted increasing attention due to their powerful ability to learn node or graph representations. However, most of the GNNs heavily rely on the assumption that the initial structure of the graph is trustworthy enough to serve as ground-truth for training. Due to uncertainty and complexity in data collection, graph structures are inevitably redundant, biased, noisy, incomplete, or the original graph structures are even unavailable, which will bring great challenges for the deployment of GNNs in real-world applications.

[*] Corresponding authors: Qiang Liu (qiang.liu@nlpr.ia.ac.cn), Jeffrey Xu Yu (yu@se.cuhk.edu.hk)

37th Conference on Neural Information Processing Systems (NeurIPS 2023) Track on Datasets and Benchmarks.

Table 1: An overview of GSLB. Both algorithms and datasets are divided into three categories: homogeneous node-level, heterogeneous node-level, and graph-level. The evaluation is divided into three dimensions: **effectiveness**, **robustness**, and **complexity**.

Algorithms	
Homogeneous GSL	LDS [11], GRCN [48], ProGNN [18], IDGL [4], CoGSL [26], SUBLIME [28], GEN [39], STABLE [21], NodeFormer [43], SLAPS [10], GSR [54], HES-GSL [42]
Heterogeneous GSL	GTN [49], HGSL [53]
Graph-level GSL	HGP-SL [52], VIB-GSL [36]
Datasets	
Homogeneous datasets	Cora [47], Citeseer [47], Pubmed [47], ogbn-arxiv [15], Polblogs, Cornell [34], Texas [34], Wisconsin [34], Actor [37]
Heterogeneous datasets	ACM [49], DBLP [49], Yelp [29]
Graph-level datasets	IMDB-B [3], IMDB-M [3], COLLAB [46], REDDIT-B [46], MUTAG [5], PROTEINS [2], Peptides-Func [8], Peptides-Struct [8]
Evaluations	
Effectiveness	Homogeneous node classification (Topology Refinement/Topology Inference), Heterogeneous node classification, Graph-level tasks
Robustness	Supervision signal robustness, Structure robustness, Feature robustness
Complexity	Time complexity, Space complexity

To mitigate the aforementioned problems, Graph Structure Learning (GSL) [4, 55, 30, 10, 57, 50] has become an important theme in graph learning. GSL aims to make the computation structure of GNNs more suitable for downstream tasks and improve the quality of the learned representations. While it is widespread in different communities and the research enthusiasm for GSL is increasing, there is no standardized benchmark that could offer a fair and consistent comparison of different GSL algorithms. Moreover, due to the complexity and diversity of graph datasets, the experimental setups in existing work are not consistent, such as varying ratios of the training set and different train/validation/test splits. This poses a great obstacle to a holistic understanding of the current research status. Therefore, the development of a standardized and comprehensive benchmark for GSL is an urgent need within the community.

In this work, we propose Graph Structure Learning Benchmark (GSLB), which serves as the first comprehensive benchmark for GSL. Our benchmark encompasses 16 state-of-the-art GSL algorithms and 20 diverse graph datasets covering homogeneous node-level, heterogeneous node-level, and graph-level tasks. We systematically investigate the characteristics of GSL in terms of three dimensions: **effectiveness**, **robustness**, and **complexity**. Based on these three dimensions, we conduct an extensive comparative study of existing GSL algorithms in different scenarios. For **effectiveness**, GSLB provides a fair and comprehensive comparison of existing algorithms on homogeneous node-level, heterogeneous node-level, and graph-level tasks, where we consider both homophilic and heterophilic graph datasets for homogeneous node-level tasks, and cover both Topology Refinement (TR, i.e., refining graphs from data with the original topology) and Topology Inference (TI, i.e., inferring graphs from data without initial topology) settings. For **robustness**, GSLB evaluates GSL models under three types of noise: supervision signal noise, structure noise, and feature noise. We also compare GSL algorithms with the models specifically designed to improve these types of robustness. For **complexity**, GSLB conducts a detailed evaluation of representative GSL algorithms in terms of time complexity and space complexity.

Through extensive experiments, we observe that: (1) GSL generally brings performance improvement for node-level tasks, especially on heterophilic graphs; (2) on graph-level tasks, current GSL models bring limited improvement and their performance varies greatly across different datasets; (3) most GSL algorithms (especially unsupervised GSL algorithms) show impressive robustness; (4) GSL models require significant time and memory overhead, making them challenging to deploy on large-scale graphs. In summary, we make the following three contributions:

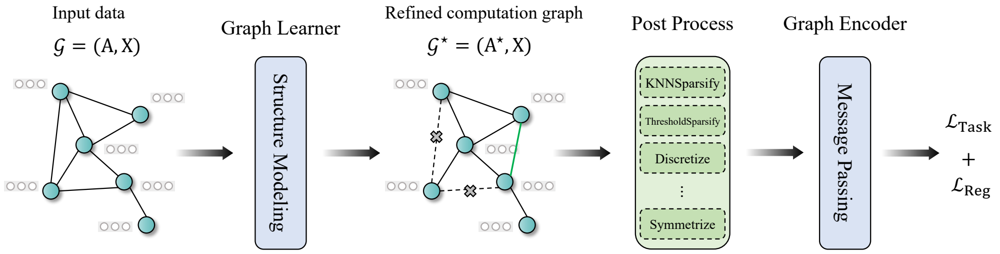

Figure 1: A general framework of Graph Structure Learning (GSL). GSL methods start with input features and an optional initial graph structure. Its corresponding computation graph is refined/inferred through a structure learning module. With the learned computation graph, Graph Neural Networks (GNNs) are used to generate graph representations.

- We propose GSLB, the first comprehensive benchmark for graph structure learning. We integrate 16 state-of-the-art GSL algorithms and 20 diverse graph datasets covering homogeneous node-level, heterogeneous node-level, and graph-level tasks. An overview of our benchmark is shown in Table 1.
- To explore the ability and limitations of GSL, we systematically evaluate existing algorithms from three dimensions: **effectiveness**, **robustness**, and **complexity**. Based on the results, we reveal the potential benefits and drawbacks of GSL to assist future research efforts.
- To facilitate future work and help researchers quickly use the latest models, we develop an easy-to-use open-source library. Besides, users can evaluate their own models or datasets with less effort. Our code is available at `https://github.com/GSL-Benchmark/GSLB`.

2 Problem Definition

In this section, we will briefly review the advances and basic concepts of GSL. Given an undirected graph $\mathcal{G} = (\mathbf{A}, \mathbf{X})$, where $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the adjacency matrix, $a_{uv} = 1$ if edge (u, v) exists and $a_{uv} = 0$ otherwise, and $\mathbf{X} \in \mathbb{R}^{N \times F}$ is the node features matrix, N is the number of nodes, F is the dimension of node features. Given an optional graph $\mathcal{G}$, the goal of GSL is to jointly optimize computation graph $\mathcal{G}^\star = (\mathbf{A}^\star, \mathbf{X})$ and the parameters of graph encoder Θ_f to obtain high-quality node representations $\mathbf{Z}^\star \in \mathbb{R}^{N \times F'}$ for downstream tasks, where $\mathbf{A}^\star$ is the refined graph by graph learner.

In general, the objective of GSL can be summarized as the following formula:

$$\mathcal{L}_{\text{GSL}} = \mathcal{L}_{\text{Task}}(\mathbf{Z}^\star, \mathbf{Y}) + \lambda \mathcal{L}_{\text{Reg}}(\mathbf{A}^\star, \mathbf{Z}^\star, \mathcal{G}) \tag{1}$$

where the first term $\mathcal{L}_{\text{Task}}$ refers to a task-specific objective with respect to the learned representation $\mathbf{Z}^\star$ and ground-truth $\mathbf{Y}$, the second term $\mathcal{L}_{\text{Reg}}$ imposes constraints on the learned graph structure and representations, and λ is a hyper-parameter that controls the trade-off between the two terms. The general framework of GSL is shown in Figure 1.

3 GSLB: Graph Structure Learning Benchmark

In this section, we introduce the overview of Graph Structure Learning Benchmark, with considerations of algorithms (Section 3.1), datasets (Section 3.2) and evaluations (Section 3.3).

3.1 Benchmark Algorithms

Table 1 shows the overall 16 algorithms integrated in GSLB. They are divided into three categories: homogeneous GSL, heterogeneous GSL, and graph-level GSL. We briefly introduce each category in the following, and more details are provided in Appendix A.2.

Homogeneous GSL. Most of the existing GSL algorithms are designed for homogeneous graphs. They assume there is only one type of nodes and edges in the graph. We select 7 TR-oriented algorithms including GRCN [48], ProGNN [18], IDGL [4], GEN [39], CoGSL [26], STABLE [21], and GSR [54]. For TI-oriented algorithms, we select SUBLIME [28], NodeFormer [43], SLAPS [10], and HES-GSL [42]. It is worth noting that TR-oriented algorithms can only be applied if the original graph structure is available, but we can construct a preliminary graph based on node features (e.g., kNN graphs or ϵ-graphs).

Heterogeneous GSL. We integrate two representative heterogeneous GSL algorithms: Graph Transformer Networks (GTN) [49] and Heterogeneous Graph Structure Learning (HGSL) [53], which can handle the heterogeneity and capture complex interactions in heterogeneous graphs.

Graph-level GSL. Graph-level GSL algorithms aim to refine each graph structure in datasets. We select two graph-level algorithms: Hierarchical Graph Pooling with Structure Learning (HGP-SL) [52] and Variational Information Bottleneck guided Graph Structure Learning (VIB-GSL) [36].

3.2 Benchmark Datasets

To comprehensively and effectively evaluate the characteristics of GSL in the field of graph learning, we have integrated a large number of datasets from various domains for different types of tasks. For node-level tasks, to evaluate the most mainstream task of GSL, node classification, we use four citation networks (i.e., Cora, Citeseer, Pubmed [47]), and ogbn-arxiv [15], three website networks from WebKB (i.e., Cornell, Texas, and Wisconsin [34]), and a cooccurrence network Actor with homophily ratio ranging from strong homophily to strong heterophily. Subsequently, to validate the effectiveness of GSL in heterogeneous node classification, we utilized three heterogeneous graph datasets (i.e., DBLP [49], ACM [49], and Yelp [29]). To investigate the robustness of GSL, we further incorporate the Polblogs dataset for evaluation. For graph-level tasks, we select six public graph classification benchmark dataset from TUDataset [31] for evaluation, including IMDB-B [3], IMDB-M [3], RDT-B [46], COLLAB [46], MUTAG [5] and PROTEINS [2]. Each dataset is a collection of graphs where each graph is associated with a level. Besides, exploring whether GSL can capture long-range information is an exciting topic. Therefore, we have utilized recently proposed long-range graph datasets: Peptides-func and Peptides-struct [8]. See more details and statistics about datasets in Appendix A.1.

3.3 Benchmark Evaluations

To comprehensively investigate the pros and cons of GSL, our benchmark evaluations encompass three dimensions: **effectiveness**, **robustness**, and **complexity**. For **effectiveness**, GSLB provides a fair and comprehensive comparison of existing algorithms from three perspectives: homogeneous node classification, heterogeneous node classification, and graph-level tasks. In the case of homogeneous node classification, we evaluated them on both homophilic and heterophilic graph datasets, conducting experiments in both TR and TI scenarios. For graph-level, we evaluate graph-level GSL algorithms on TUDataset and long-range graph datasets for exploring the capabilities on graph-level tasks. For most datasets, we use accuracy as our evaluation metric. For **robustness**, GSLB evaluates three types of robustness: supervision signal robustness, structure robustness, and feature robustness. We control the count of labels to explore the supervision signal robustness of GSL and find that GSL exhibits excellent performance in the scenarios with few labels. We inject random structure noise and graph topology attacks to investigate the structure robustness. We also study the feature robustness by randomly masking a certain proportion of node features. For **complexity**, we conduct a detailed evaluation of representative GSL algorithms in terms of time complexity and space complexity. It will help to facilitate the deployment of GSL in real-world applications.

4 Experiments and Analysis

In this section, we systematically investigate the **effectiveness**, **robustness**, and **complexity** of GSL algorithms by answering the following specific questions:

- For **effectiveness**, **RQ1**: How effective are the algorithms on node-level representation learning (Section 4.2)? **RQ2**: Can GSL mitigate homophily inductive bias of traditional message-passing based GNNs (Section 4.2)? **RQ3**: How does GSL perform on heterogeneous graph datasets (Section 4.3)? **RQ4**: How effective are the algorithms on graph-level representation learning (Section 4.4)? **RQ5**: Can GSL methods capture long-range information on the graph (Appendix B)?

- For **robustness**, **RQ6**: How robust are GSL algorithms when faced with a scarcity of labeled samples? **RQ7**: How robust are GSL algorithms in the face of structure attack or noise? **RQ8**: How is the feature robustness of GSL? (Section 4.5)
- For **complexity**, **RQ9**: How efficient are these algorithms in terms of time and space (Section 4.6)?
- Otherwise, **RQ10**: What does the learned graph structure look like (Appendix B.2)?

4.1 Experimental Settings

All algorithms in GSLB are implemented by PyTorch [33], and unless specifically indicated, the encoders for all algorithms are Graph Convolutional Networks. All experiments are conducted on a Linux server with GPU (NVIDIA GeForce 3090 and NVIDIA A100) and CPU (AMD EPYC 7763), using PyTorch 1.13.0, DGL 1.1.0 [38] and Python 3.9.16.

4.2 Performance on node-level representation learning

For node-level representation learning, we conduct experiments on homogeneous graph datasets under both TR and TI scenarios, and use classification accuracy as our evaluation metric. Table 2 shows the experimental results of various GSL algorithms under the standard setting of transductive node classification task in the TR scenario. We can observe that: 1) Most GSL algorithms generally show improvements in node classification task, particularly on datasets with high heterophily ratio. Due to the presence of heterophilic connections in heterophily graphs, where nodes are often connected to nodes with different labels, it violates the homophily assumption of message-passing neural networks. As a result, traditional GNNs like GCN and GAT exhibit poor performance. However, GSL can improve significantly on heterophily graph datasets by learning new graph structures based on downstream tasks and specific learning objectives, thus enhancing the homophily of the graph and promoting the performance on node-level representation learning. 2) SUBLIME achieves optimal or near-optimal results on most datasets. It learns graph structure through contrastive learning in an unsupervised manner. As mentioned in the recent literature [10], optimizing graph structures solely based on label information is insufficient. Leveraging a large and abundant amount of unlabeled information can enhance the performance of GSL. 3) The scalability of GSL still needs improvement, as only a few models can be trained on large-scale datasets (e.g., ogbn-arxiv). We will discuss the scalability of GSL algorithms in detail in a subsequent section (Section 4.6).

Table 3 shows the experimental results of the transductive node classification task in the TI scenario. Some GSL algorithms are designed for TR scenario (i.e., GRCN, IDGL, etc.), so we use kNN graphs as their original graph structure. As we can observe, on the homophily graph datasets, GSL outperforms baselines, such as MLP, GCN$_{knn}$ and GAT$_{knn}$. However, on the heterophily graph datasets, most GSL algorithms often have difficulty achieving better results than baseline models. As mentioned in earlier literature, a network with randomness tends to get better performance utilizing kNN for direct information propagation [17]. Therefore, traditional message-passing neural networks with kNN graphs demonstrate powerful performance. In addition, as observed in the TR scenario, models that leverage self-supervision to extract abundant unlabeled information often achieve better performance.

4.3 Performance on heterogeneous graph node-level representation learning

In this section, we evaluate the performance of GSL algorithms on heterogeneous node classification task and use Macro-F1 and Micro-F1 as our evaluation metrics. Table 4 shows the experimental results on heterogeneous graph datasets. By observing the results, we can find that: 1) Because GTN and HGSL consider both heterogeneity and structure learning, they generally outperform other models on heterogeneous graph datasets. 2) GSL algorithms generally outperform the vanilla GNN models (e.g. GCN and GAT) since they have learned better structures to facilitate message passing. 3) Due to the majority of GSL algorithms not explicitly accounting for heterogeneity, they may exhibit poorer performance on heterogeneous graph datasets. 4) Some datasets (e.g. Yelp) exhibit stronger heterogeneity, and on such datasets, models that consider heterogeneity (e.g. HAN, GTN, and HGSL) perform significantly better.

Table 2: Accuracy ± STD comparison (%) under the standard setting of transductive node classification task in the Topology Refinement (TR) scenario, which means the original graph structure is available for each method. Performance is averaged from 10 independent repetitions. The highest results are highlighted with **bold**, while the second highest results are marked with underline. "OOM" denotes out of memory.

Edge Hom.	Cora 0.81	Citeseer 0.74	Pubmed 0.80	ogbn-arxiv 0.65	Cornell 0.12	Texas 0.06	Wisconsin 0.18	Actor 0.22
GCN	81.46±0.58	71.36±0.31	79.18±0.29	70.77±0.19	47.84±5.55	57.83±2.76	57.45±4.30	30.01±0.77
GAT	81.41±0.77	70.69±0.58	77.85±0.42	69.90±0.25	46.22±6.33	54.05±7.35	57.65±7.75	28.91±0.83
GPRGNN	83.66±0.77	71.64±0.49	75.99±1.63	50.80±0.29	**76.76±5.30**	**85.14±3.68**	**83.33±3.42**	**34.09±1.09**
LDS	83.01±0.41	**73.55±0.54**	OOM	OOM	47.87±7.14	58.92±4.32	61.70±3.58	31.05±1.31
GRCN	83.87±0.49	72.43±0.61	78.92±0.39	OOM	54.32±8.24	62.16±7.05	56.08±7.19	29.97±0.71
ProGNN	80.30±0.57	68.51±0.52	OOM	OOM	54.05±6.16	48.37±12.17	62.54±7.56	22.35±0.88
IDGL	**83.88±0.42**	72.20±1.18	80.00±0.38	OOM	50.00±8.98	62.43±6.09	59.41±4.11	28.16±1.41
GEN	80.21±1.72	71.15±1.81	78.91±0.69	OOM	57.02±7.19	65.94±1.38	66.07±3.72	27.21±2.05
CoGSL	81.76±0.24	73.09±0.42	OOM	OOM	52.16±3.21	59.46±4.36	58.82±1.52	32.95±1.20
SUBLIME	83.40±0.42	72.30±1.09	**80.90±0.94**	**71.75±0.36**	70.54±5.98	77.03±4.23	78.82±6.55	33.57±0.68
STABLE	80.20±0.68	68.91±1.01	OOM	OOM	44.03±4.05	55.24±6.04	53.00±5.27	30.18±1.00
NodeFormer	80.28±0.82	71.31±0.98	78.21±1.43	55.40±0.23	42.70±5.51	58.92±4.32	48.43±7.02	25.51±1.77
GSR	82.48±0.43	71.10±0.25	78.09±0.53	OOM	44.32±2.16	60.81±4.87	56.86±1.24	30.23±0.38

Table 3: Accuracy ± STD comparison (%) under the standard setting of transductive node classification task in the Topology Inference (TI) scenario, which means the original graph structure is not available for each method.

Edge Hom.	Cora 0.81	Citeseer 0.74	Pubmed 0.80	ogbn-arxiv 0.65	Cornell 0.12	Texas 0.06	Wisconsin 0.18	Actor 0.22
MLP	58.55±0.80	59.52±0.64	73.00±0.30	55.21±0.11	71.35±6.19	80.27±5.93	**84.71±3.14**	35.49±1.04
GCN$_{knn}$	66.10±0.44	68.33±0.89	69.23±0.49	55.21±0.22	75.14±2.65	75.95±4.43	84.12±3.97	32.98±0.49
GAT$_{knn}$	64.62±1.04	68.05±1.12	68.76±0.80	55.92±0.30	74.05±5.16	76.49±4.99	82.16±4.06	31.67±1.19
GPRGNN$_{knn}$	69.27±0.62	70.29±0.54	68.19±1.19	51.39±0.13	**75.68±2.70**	**81.08±4.18**	84.12±3.22	34.71±1.51
LDS	69.87±0.41	72.43±0.61	OOM	OOM	72.65±3.86	70.20±5.07	78.14±4.50	32.39±0.79
GRCN$_{knn}$	69.48±0.66	68.41±0.50	68.96±0.85	OOM	71.08±6.84	74.32±5.02	78.63±4.92	30.83±0.76
ProGNN$_{knn}$	67.11±0.56	64.55±0.95	OOM	OOM	71.35±4.04	71.89±5.69	72.94±7.93	31.56±1.14
IDGL$_{knn}$	69.74±0.57	66.33±0.84	74.01±0.64	OOM	72.70±4.75	75.40±4.75	79.21±3.94	33.07±1.37
GEN$_{knn}$	66.95±1.40	67.29±1.17	69.76±1.53	OOM	71.08±5.54	74.59±3.46	81.76±2.91	31.28±1.06
CoGSL$_{knn}$	66.65±0.37	68.72±0.84	OOM	OOM	70.27±3.42	72.70±4.26	76.96±5.25	34.52±1.56
GSR$_{knn}$	66.28±0.59	66.77±0.62	68.49±1.49	OOM	70.27±3.62	74.86±3.63	78.62±5.91	33.73±1.12
SLAPS	72.28±0.97	70.71±1.13	74.50±1.47	55.19±0.21	74.59±3.67	79.19±4.99	81.96±3.26	**37.16±0.91**
SUBLIME	72.74±1.91	**72.63±0.60**	75.08±0.55	55.57±0.18	72.35±3.57	75.51±5.08	82.14±2.62	32.20±1.02
NodeFormer	54.35±5.33	45.90±5.42	59.83±6.50	55.37±0.23	42.70±5.51	58.92±4.32	48.24±6.63	29.24±1.68
HES-GSL	**73.68±1.04**	70.12±1.11	**77.08±0.78**	**56.46±0.27**	66.22±6.19	74.05±6.42	79.61±5.28	36.73±0.76

4.4 Performance of GSL algorithms on graph-level tasks

In this section, we conduct graph classification experiments on four social datasets (i.e., `IMDB-B`, `RDT-B`, `COLLAB`, and `IMDB-M`) and two biological datasets (i.e., `MUTAG` and `PROTEINS`). Table 5 shows the experimental results of average accuracy and the standard deviation of 10-fold cross-validation. We can observe that HGP-SL (with GCN as the encoder) consistently outperforms GCN on all datasets. However, we find that VIB-GSL exhibits strong instability across different random seeds. And due to the absence of training scripts in the official code[2], we performed hyperparameter tuning based on the parameter search space ($\beta \in \{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\}$) provided in the paper, but we are unable to surpass the performance of the baseline models consistently. Lastly, we conducted an analysis of graph-level GSL algorithms on long-range graph dataset [8]. For detailed information, please refer to Appendix B.

[2] `https://github.com/RingBDStack/VIB-GSL`

Table 4: Macro-F1 and Micro-F1 ± STD comparison (%) under the standard setting of heterogeneous node classification task.

Method	ACM		DBLP		Yelp	
	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1
GCN	90.27±0.59	90.18±0.61	90.01±0.32	90.99±0.28	78.01±1.89	81.03±1.81
GAT	91.52±0.62	91.46±0.62	90.22±0.37	91.13±0.40	82.12±1.47	84.43±1.56
HAN	91.67±0.39	91.47±0.22	90.53±0.24	91.47±0.22	88.49±1.73	88.78±1.40
LDS	92.35±0.43	92.05±0.26	88.11±0.86	88.74±0.85	75.98±2.35	78.14±1.98
GRCN	93.04±0.17	92.94±0.18	88.33±0.47	89.43±0.44	76.05±1.05	80.68±0.96
IDGL	91.69±1.24	91.63±1.24	89.65±0.60	90.61±0.56	76.98±5.78	79.15±5.06
ProGNN	90.57±1.03	90.50±1.29	83.13±1.56	84.83±1.36	51.76±1.46	58.39±1.25
GEN	87.91±2.78	87.88±2.61	89.74±0.69	90.65±0.71	80.43±3.78	82.68±2.84
STABLE	83.54±4.20	83.38±4.51	75.18±1.95	76.42±1.95	71.48±4.71	76.62±2.75
GEN	87.91±2.78	87.88±2.61	89.74±0.69	90.65±0.71	80.43±3.78	82.68±2.84
SUBLIME	92.42±0.16	92.13±0.37	90.98±0.37	91.82±0.27	79.68±0.79	82.99±0.82
NodeFormer	91.33±0.77	90.60±0.95	79.54±0.78	80.56±0.62	91.69±0.65	90.59±1.21
GSR	92.14±1.08	92.11±0.99	76.59±0.45	77.69±0.42	83.85±0.76	85.73±0.54
GTN	92.04±0.38	91.94±0.39	90.52±0.45	91.48±0.39	**92.98±0.52**	**92.44±0.46**
HGSL	**93.23±0.50**	**93.13±0.51**	**91.58±0.40**	**92.49±0.35**	92.79±0.44	92.24±0.48

Table 5: Accuracy ± STD comparison (%) under the setting of graph-level classification task.

Method	IMDB-B	RDT-B	COLLAB	IMDB-M	MUTAG	PROTEINS
GCN	73.20±4.29	70.10±5.80	76.96±2.28	49.85±3.84	73.92±8.84	67.52±6.71
VIB-GSL (GCN)	71.90±4.48	68.95±2.66	77.14±1.59	49.05±5.52	68.63±5.15	65.68±8.53
HGP-SL (GCN)	**74.10±4.55**	OOM	78.06±2.17	**51.07±2.00**	78.07±10.85	70.80±4.25
GAT	72.30±2.26	73.55±4.76	79.08±1.36	48.90±2.98	78.71±7.51	68.63±6.24
VIB-GSL (GAT)	72.10±5.69	OOM	77.54±1.85	49.06±4.55	77.13±9.95	67.09±8.43
SAGE	72.60±3.69	70.20±4.11	75.58±2.04	48.55±2.03	68.65±4.31	64.47±7.15
VIB-GSL (SAGE)	73.00±4.78	65.75±3.17	77.74±1.52	48.79±5.06	72.81±11.41	66.61±4.48
HGP-SL (SAGE)	71.50±5.24	OOM	78.64±1.47	49.67±3.09	77.13±3.29	73.32±2.06
GIN	73.00±2.67	71.70±5.01	79.86±1.64	50.30±3.52	**87.19±8.05**	69.07±5.62
VIB-GSL (GIN)	69.90±3.90	**75.85±3.63**	77.25±2.34	49.97±3.65	85.18±10.11	**75.15±5.72**
HGP-SL (GIN)	73.50±6.25	OOM	**80.14±1.51**	48.67±2.58	73.92±6.24	69.37±3.95

4.5 Robustness analysis of GSL algorithms

To investigate the robustness of GSL algorithms, we primarily focus on three aspects: structure robustness, feature robustness, and supervision signal robustness. Due to limited space, we predominantly investigate the transductive node classification task in our paper. Nevertheless, researchers can utilize our GSLB library to efficiently and conveniently conduct experiments on other tasks as well.

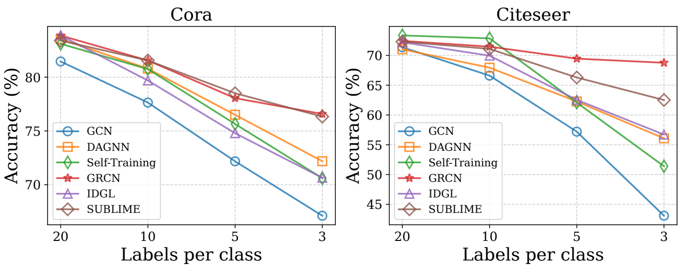

Figure 2: Performance of baselines and GSL algorithms with respect to different numbers of labels per class on Cora and Citeseer.

Robustness analysis with respect to different supervision signals. We have discovered that GSL maintains surprising performance in scenarios with a scarcity of labeled samples. We varied the number of labels per class in the Cora and Citeseer datasets and selected two baseline models, DAGNN [25] and Self-Training [22], that performed well in scenarios with limited labels. As shown in Fig-

Table 6: Accuracy ± STD comparison (%) with respect to different perturbation rates. Jaccard and SimPGCN are representative state-of-the-art defense GNNs.

Dataset	Ptb Rate	GCN	Jaccard	SimPGCN	IDGL	GRCN	ProGNN	STABLE	SUBLIME
Cora	0%	83.68±0.37	83.78±0.50	82.66±0.48	**84.69±1.13**	84.43±0.26	84.53±0.89	83.70±0.30	83.84±0.28
	5%	80.61±0.39	81.44±0.48	80.35±0.82	**82.56±0.24**	81.34±0.50	81.47±0.44	81.52±0.85	79.93±0.58
	10%	74.38±0.59	75.90±0.64	76.50±1.12	78.06±0.62	77.12±0.38	72.61±0.73	78.64±1.82	**78.71±0.46**
	15%	65.17±0.99	77.14±0.70	73.77±1.88	76.88±0.44	73.74±0.61	65.68±1.97	**79.70±1.71**	79.34±0.61
	20%	61.98±1.23	70.71±0.91	69.08±2.78	67.19±0.69	69.54±0.58	61.07±0.61	**76.44±2.47**	75.25±1.08
Citeseer	0%	**76.56±0.36**	74.34±0.26	74.35±0.74	73.87±0.70	76.34±0.11	73.36±1.52	72.65±1.36	73.34±1.17
	5%	72.51±0.30	70.01±0.79	72.99±1.05	72.46±0.47	**74.66±0.27**	71.46±0.47	69.66±0.95	72.63±0.50
	10%	71.92±0.68	70.28±1.30	72.68±0.54	69.72±0.59	**74.06±0.43**	69.03±0.60	72.79±0.71	73.02±0.29
	15%	64.44±0.53	67.13±1.28	71.74±1.46	62.83±1.28	66.46±1.12	65.42±1.20	70.98±0.61	**73.90±0.52**
	20%	57.51±1.03	67.82±0.74	70.06±1.86	61.16±0.99	69.42±1.14	57.51±0.36	71.90±1.12	**72.55±0.62**
Polblogs	0%	95.62±0.69	94.93±0.28	94.50±0.43	94.83±0.20	**95.65±0.28**	94.84±0.19	95.63±0.32	95.27±0.51
	5%	80.57±0.66	78.17±0.55	76.02±1.14	79.62±0.65	**93.70±0.18**	92.36±0.42	89.41±1.63	93.24±1.50
	10%	71.83±2.37	71.86±1.34	70.12±1.10	74.54±0.69	87.99±1.56	84.66±0.52	89.87±0.82	**93.62±0.50**
	15%	66.38±2.17	69.93±0.66	64.19±1.55	75.53±0.83	71.85±1.58	77.38±0.51	89.94±0.89	**94.29±0.27**
	20%	68.19±2.24	69.22±0.34	63.64±1.41	71.63±0.62	71.73±1.58	73.57±0.29	87.42±0.69	**92.60±0.72**

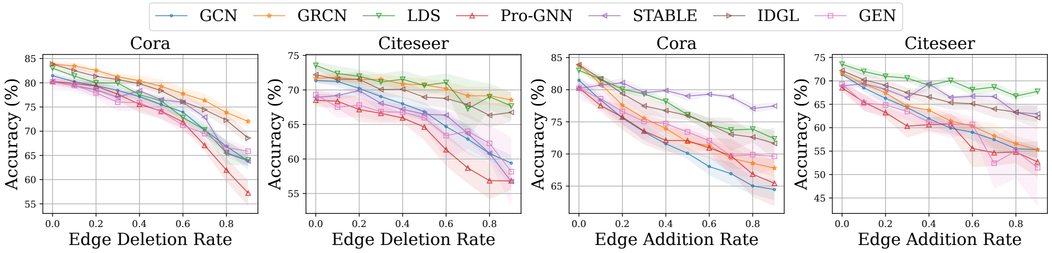

Figure 3: Analysis of robustness when injecting random noise on Cora and Citeseer.

ure 2, we can observe that GSL algorithms (for the sake of brevity, we opted to select only three models: GRCN, IDGL, and SUBLIME) achieve the best results in scenarios with fewer labels available. We speculate that this may be because the learned graph structure is denser and exhibits cleaner community boundaries. As a result, the supervision signals can propagate more effectively within such a structure.

Robustness analysis with respect to random noise. We randomly remove edges from or add edges to the original graph structures of `Cora` and `Citeseer`, then evaluated the performance of GSL algorithms on the corrupted graphs. We change the ratios of modified edges from 0 to 0.9 to simulate different attack intensities. As shown in Figure 3, as the noise intensity increases, the models' performance generally exhibits a downward trend. And we can observe that GSL algorithms commonly demonstrate a certain degree of robustness, as they tend to exhibit more stable performance than GCN when random noise is injected. Besides, we also found that, due to variations in the graph modeling process, different algorithms display varying levels of robustness when facing edge deletion and edge addition scenarios. For example, GRCN demonstrates strong robustness in edge deletion scenarios. However, in the edge addition scenarios, it only exhibits slight performance improvements compared to GCN. On the contrary, STABLE exhibits strong robustness in the edge deletion scenario, while showing the opposite trend in edge addition.

Robust analysis with respect to graph topology attack. Following [21, 55], we conduct robust analysis on three graph datasets, i.e., Cora, Citeseer, and Polblogs. First, we select the largest connected component in the graph, and utilize Mettack [58], a non-targeted adversarial topology attack method, to generate perturbed graphs. We select the perturbation rate from 0% to 20%. Table 6 shows the performance of GSL algorithms on three datasets with respect to various perturbation rates. Surprisingly, we can observe that most GSL algorithms exhibit strong robustness against graph topology attacks, even better than state-of-the-art defense GNNs (e.g., Jaccard [41] and SimPGCN [19]). GSL can effectively remove the newly added adversarial edges, and recover important edges to promote message passing. As mentioned in Li et al. [21], optimizing graph structures based on either features or supervised signals might not be reliable. We found that self-supervised graph structure modeling methods (e.g., STABLE and SUBLIME) show excellent

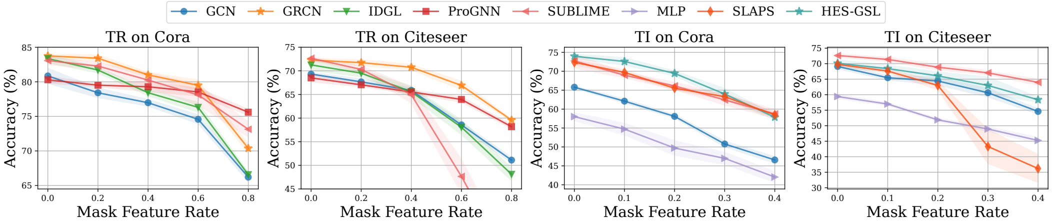

Figure 4: Analysis of robustness when injecting random feature noise on Cora and Citeseer.

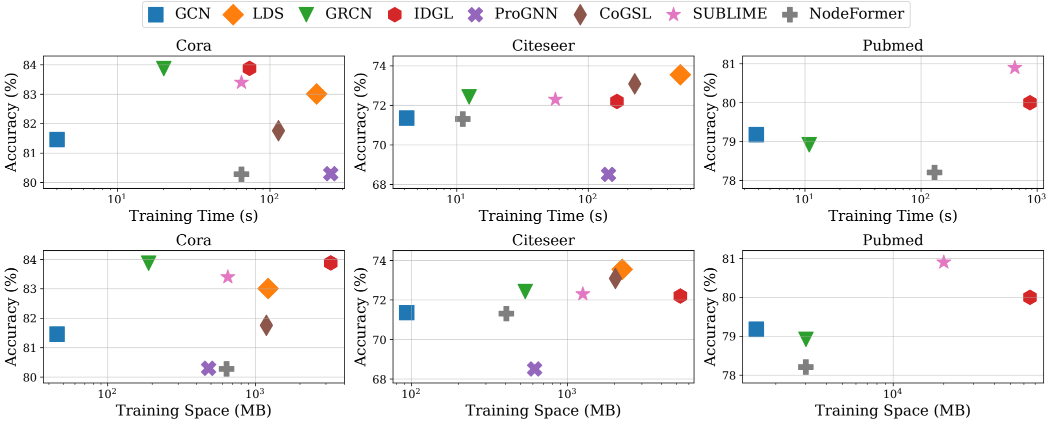

Figure 5: Training time and space analysis on `Cora`, `Citeseer` and `Pubmed`.

performance on corrupted graph structure datasets, which means unsupervised representation learning might produce more reliable and high-quality representations to conduct structure modeling.

Robust analysis with respect to feature noise. On the basis of exploring structural robustness, we also study the feature robustness of GSL. We randomly mask a certain proportion of node features by filling them with zeros, to investigate the performance of GSL algorithms when node features are subjected to varying degrees of damage. As shown in Figure 4, we can observe that: 1) the node features play a more critical role than the structure on certain datasets. Under the same noise degree, feature noise brings more performance degradation compared with structure noise; 2) Interestingly, while most existing GSL methods rely on feature similarity between pairs of nodes to learn graph structure, they still exhibit good robustness when facing noisy node features; 3) edge-oriented algorithms (e.g., ProGNN) show stronger feature robustness, because they optimize adjacency matrix directly, and have less dependence on pairs of node features.

4.6 Efficiency and scalability analysis

In this section, we analyze the efficiency and scalability of GSL algorithms on Cora, Citeseer, and Pubmed datasets. For time efficiency, we evaluate the efficiency of the algorithms by measuring the time it takes for them to converge, i.e., achieve the best performance on the validation set. For scalability, we set all models to their dense version to ensure a fair comparison. As shown in Figure 5, GSL algorithms generally have higher time and space complexity compared to GCN. This limitation restricts the application of GSL on large-scale graphs. We can observe that some algorithms (e.g., GRCN) can achieve relatively good performance improvement with less complexity increase. Besides, although some algorithms (e.g., IDGL, LDS, and SUBLIME) achieve remarkable effectiveness improvement, they largely increase the complexity of time and space.

5 Conclusion and Future Directions

In this paper, we give a brief introduction and overview of graph structure learning. Then we present the first Graph Structure Learning Benchmark (GSLB) consisting of 16 algorithms and 20 datasets for various tasks. Based on GSLB, we conducted extensive experiments to reveal and analyze the performance of GSL algorithms in different scenarios and tasks. Through our comparative study, we find that GSL achieves promising results in heterophily, robustness, etc. The goal of this work is to understand the current state of development of GSL and provide insights for future research.

Notwithstanding the promising results that have been made, there are still some critical challenges and research directions worthy of future investigation.

- **Insufficient scalability**. Most existing works model the existence probability of edges based on node pairs, with a complexity of $O(N^2)$. This makes it challenging to employ GSL in large-scale graphs in real-world applications. Future work should focus on overcoming the limitations of GSL in terms of complexity.
- **Surprising performance with few labels**. We have observed that GSL learns denser and more distinct graph structures, which facilitates the propagation of supervision signals. Most existing GNNs that address few label problem are based on deep GNNs [25, 13] or semi-supervised approaches [6, 35, 20], without refining the graph structure. In the future, it would be worth exploring the combination of increasing the supervision signals and making the graph structure more suitable for propagating those signals.
- **Excellent performance of unsupervised GSL in robustness**. Some algorithms using self-supervised methods for learning graph structures exhibit excellent performance in robustness, which may be attributed to the avoidance of unreliable supervision signals. In the future, further exploration can be done to utilize unsupervised structure learning for designing defense models.
- **Hard to apply on incomplete graphs.** Most existing algorithms rely on pairwise node embeddings to generate the probability of edge existence. The underlying assumption is that all attributes of nodes on the graph are complete. However, it is common in practice that some nodes or all nodes have no features. Future research should address the challenges of structure learning on incomplete graphs.

Acknowledge

This work was partially supported by the Research Grants Council of Hong Kong, No. 14202919 and No. 14205520.

References

[1] Tian Bian, Xi Xiao, Tingyang Xu, Peilin Zhao, Wenbing Huang, Yu Rong, and Junzhou Huang. Rumor detection on social media with bi-directional graph convolutional networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 549–556, 2020. 1

[2] Karsten M Borgwardt, Cheng Soon Ong, Stefan Schönauer, SVN Vishwanathan, Alex J Smola, and Hans-Peter Kriegel. Protein function prediction via graph kernels. *Bioinformatics*, 21(suppl_1):i47–i56, 2005. 2, 4

[3] Chen Cai and Yusu Wang. A simple yet effective baseline for non-attributed graph classification. *arXiv preprint arXiv:1811.03508*, 2018. 2, 4

[4] Yu Chen, Lingfei Wu, and Mohammed Zaki. Iterative deep graph learning for graph neural networks: Better and robust node embeddings. *Advances in neural information processing systems*, 33:19314–19326, 2020. 2, 4, 16

[5] Asim Kumar Debnath, Rosa L Lopez de Compadre, Gargi Debnath, Alan J Shusterman, and Corwin Hansch. Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity. *Journal of medicinal chemistry*, 34(2): 786–797, 1991. 2, 4

[6] Kaize Ding, Jianling Wang, James Caverlee, and Huan Liu. Meta propagation networks for graph few-shot semi-supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 6524–6531, 2022. 10

[7] Paul D Dobson and Andrew J Doig. Distinguishing enzyme structures from non-enzymes without alignments. *Journal of molecular biology*, 330(4):771–783, 2003. 15

[8] Vijay Prakash Dwivedi, Ladislav Rampášek, Michael Galkin, Ali Parviz, Guy Wolf, Anh Tuan Luu, and Dominique Beaini. Long range graph benchmark. *Advances in Neural Information Processing Systems*, 35:22326–22340, 2022. 2, 4, 6, 15, 17

[9] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. Graph neural networks for social recommendation. In *The world wide web conference*, pages 417–426, 2019. 1

[10] Bahare Fatemi, Layla El Asri, and Seyed Mehran Kazemi. Slaps: Self-supervision improves structure learning for graph neural networks. *Advances in Neural Information Processing Systems*, 34:22667–22681, 2021. 2, 4, 5, 16

[11] Luca Franceschi, Mathias Niepert, Massimiliano Pontil, and Xiao He. Learning discrete structures for graph neural networks. In *International conference on machine learning*, pages 1972–1982. PMLR, 2019. 2, 15, 16

[12] Xiang Gao, Wei Hu, and Zongming Guo. Exploring structure-adaptive graph learning for robust semi-supervised classification. In *2020 ieee international conference on multimedia and expo (icme)*, pages 1–6. IEEE, 2020. 16

[13] Johannes Gasteiger, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. *arXiv preprint arXiv:1810.05997*, 2018. 10

[14] Zhongkai Hao, Chengqiang Lu, Zhenya Huang, Hao Wang, Zheyuan Hu, Qi Liu, Enhong Chen, and Cheekong Lee. Asgn: An active semi-supervised graph neural network for molecular property prediction. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 731–752, 2020. 1

[15] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133, 2020. 2, 4, 14

[16] Bo Jiang, Ziyan Zhang, Doudou Lin, Jin Tang, and Bin Luo. Semi-supervised learning with graph learning-convolutional networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11313–11320, 2019. 16

[17] Di Jin, Zhizhi Yu, Cuiying Huo, Rui Wang, Xiao Wang, Dongxiao He, and Jiawei Han. Universal graph convolutional networks. *Advances in Neural Information Processing Systems*, 34:10654–10664, 2021. 5

[18] Wei Jin, Yao Ma, Xiaorui Liu, Xianfeng Tang, Suhang Wang, and Jiliang Tang. Graph structure learning for robust graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 66–74, 2020. 2, 4, 15, 16

[19] Wei Jin, Tyler Derr, Yiqi Wang, Yao Ma, Zitao Liu, and Jiliang Tang. Node similarity preserving graph convolutional networks. In *Proceedings of the 14th ACM international conference on web search and data mining*, pages 148–156, 2021. 8

[20] Junseok Lee, Yunhak Oh, Yeonjun In, Namkyeong Lee, Dongmin Hyun, and Chanyoung Park. Grafn: Semi-supervised node classification on graph with few labels via non-parametric distribution assignment. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2243–2248, 2022. 10

[21] Kuan Li, Yang Liu, Xiang Ao, Jianfeng Chi, Jinghua Feng, Hao Yang, and Qing He. Reliable representations make a stronger defender: Unsupervised structure refinement for robust gnn. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 925–935, 2022. 2, 4, 8, 16

[22] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018. 7

[23] Zhixun Li, Dingshuo Chen, Qiang Liu, and Shu Wu. The devil is in the conflict: Disentangled information graph neural networks for fraud detection. *arXiv preprint arXiv:2210.12384*, 2022. 1

[24] Derek Lim, Felix Hohne, Xiuyu Li, Sijia Linda Huang, Vaishnavi Gupta, Omkar Bhalerao, and Ser Nam Lim. Large scale learning on non-homophilous graphs: New benchmarks and strong simple methods. *Advances in Neural Information Processing Systems*, 34:20887–20902, 2021. 17

[25] Meng Liu, Hongyang Gao, and Shuiwang Ji. Towards deeper graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 338–348, 2020. 7, 10

[26] Nian Liu, Xiao Wang, Lingfei Wu, Yu Chen, Xiaojie Guo, and Chuan Shi. Compact graph structure learning via mutual information compression. In *Proceedings of the ACM Web Conference 2022*, pages 1601–1610, 2022. 2, 4, 16

[27] Yang Liu, Xiang Ao, Zidi Qin, Jianfeng Chi, Jinghua Feng, Hao Yang, and Qing He. Pick and choose: a gnn-based imbalanced learning approach for fraud detection. In *Proceedings of the Web Conference 2021*, pages 3168–3177, 2021. 1

[28] Yixin Liu, Yu Zheng, Daokun Zhang, Hongxu Chen, Hao Peng, and Shirui Pan. Towards unsupervised deep graph structure learning. In *Proceedings of the ACM Web Conference 2022*, pages 1392–1403, 2022. 2, 4, 16

[29] Yuanfu Lu, Chuan Shi, Linmei Hu, and Zhiyuan Liu. Relation structure-aware heterogeneous information network embedding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4456–4463, 2019. 2, 4, 14

[30] Dongsheng Luo, Wei Cheng, Wenchao Yu, Bo Zong, Jingchao Ni, Haifeng Chen, and Xiang Zhang. Learning to drop: Robust graph neural network via topological denoising. In *Proceedings of the 14th ACM international conference on web search and data mining*, pages 779–787, 2021. 2, 16

[31] Christopher Morris, Nils M Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. Tudataset: A collection of benchmark datasets for learning with graphs. *arXiv preprint arXiv:2007.08663*, 2020. 4

[32] Mark EJ Newman. Mixing patterns in networks. *Physical review E*, 67(2):026126, 2003. 18

[33] 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, 2019. 5

[34] Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks. *arXiv preprint arXiv:2002.05287*, 2020. 2, 4, 14, 17

[35] Ke Sun, Zhouchen Lin, and Zhanxing Zhu. Multi-stage self-supervised learning for graph convolutional networks on graphs with few labeled nodes. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 5892–5899, 2020. 10

[36] Qingyun Sun, Jianxin Li, Hao Peng, Jia Wu, Xingcheng Fu, Cheng Ji, and S Yu Philip. Graph structure learning with variational information bottleneck. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 4165–4174, 2022. 2, 4, 16, 17

[37] Jie Tang, Jimeng Sun, Chi Wang, and Zi Yang. Social influence analysis in large-scale networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 807–816, 2009. 2, 14

[38] Minjie Yu Wang. Deep graph library: Towards efficient and scalable deep learning on graphs. In *ICLR workshop on representation learning on graphs and manifolds*, 2019. 5

[39] Ruijia Wang, Shuai Mou, Xiao Wang, Wanpeng Xiao, Qi Ju, Chuan Shi, and Xing Xie. Graph structure estimation neural networks. In *Proceedings of the Web Conference 2021*, pages 342–353, 2021. 2, 4, 16

[40] Oliver Wieder, Stefan Kohlbacher, Mélaine Kuenemann, Arthur Garon, Pierre Ducrot, Thomas Seidel, and Thierry Langer. A compact review of molecular property prediction with graph neural networks. *Drug Discovery Today: Technologies*, 37:1–12, 2020. 1

[41] Huijun Wu, Chen Wang, Yuriy Tyshetskiy, Andrew Docherty, Kai Lu, and Liming Zhu. Adversarial examples on graph data: Deep insights into attack and defense. *arXiv preprint arXiv:1903.01610*, 2019. 8

[42] Lirong Wu, Haitao Lin, Zihan Liu, Zicheng Liu, Yufei Huang, and Stan Z Li. Homophily-enhanced self-supervision for graph structure learning: Insights and directions. *IEEE Transactions on Neural Networks and Learning Systems*, 2023. 2, 4, 16

[43] Qitian Wu, Wentao Zhao, Zenan Li, David P Wipf, and Junchi Yan. Nodeformer: A scalable graph structure learning transformer for node classification. *Advances in Neural Information Processing Systems*, 35:27387–27401, 2022. 2, 4, 16

[44] Hui Xu, Liyao Xiang, Jiahao Yu, Anqi Cao, and Xinbing Wang. Speedup robust graph structure learning with low-rank information. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 2241–2250, 2021. 16

[45] Weizhi Xu, Junfei Wu, Qiang Liu, Shu Wu, and Liang Wang. Evidence-aware fake news detection with graph neural networks. In *Proceedings of the ACM Web Conference 2022*, pages 2501–2510, 2022. 1

[46] Pinar Yanardag and SVN Vishwanathan. Deep graph kernels. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1365–1374, 2015. 2, 4

[47] Zhilin Yang, William Cohen, and Ruslan Salakhudinov. Revisiting semi-supervised learning with graph embeddings. In *International conference on machine learning*, pages 40–48. PMLR, 2016. 2, 4, 14

[48] Donghan Yu, Ruohong Zhang, Zhengbao Jiang, Yuexin Wu, and Yiming Yang. Graph-revised convolutional network. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part III*, pages 378–393. Springer, 2021. 2, 4, 15, 16

[49] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. Graph transformer networks. *Advances in neural information processing systems*, 32, 2019. 2, 4, 14, 16, 17

[50] Jinghao Zhang, Yanqiao Zhu, Qiang Liu, Shu Wu, Shuhui Wang, and Liang Wang. Mining latent structures for multimedia recommendation. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 3872–3880, 2021. 2

[51] Yanfu Zhang, Hongchang Gao, Jian Pei, and Heng Huang. Robust self-supervised structural graph neural network for social network prediction. In *Proceedings of the ACM Web Conference 2022*, pages 1352–1361, 2022. 1

[52] Zhen Zhang, Jiajun Bu, Martin Ester, Jianfeng Zhang, Chengwei Yao, Zhi Yu, and Can Wang. Hierarchical graph pooling with structure learning. *arXiv preprint arXiv:1911.05954*, 2019. 2, 4, 16, 17

[53] Jianan Zhao, Xiao Wang, Chuan Shi, Binbin Hu, Guojie Song, and Yanfang Ye. Heterogeneous graph structure learning for graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 4697–4705, 2021. 2, 4, 16, 17

[54] Jianan Zhao, Qianlong Wen, Mingxuan Ju, Chuxu Zhang, and Yanfang Ye. Self-supervised graph structure refinement for graph neural networks. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pages 159–167, 2023. 2, 4, 16

[55] Cheng Zheng, Bo Zong, Wei Cheng, Dongjin Song, Jingchao Ni, Wenchao Yu, Haifeng Chen, and Wei Wang. Robust graph representation learning via neural sparsification. In *International Conference on Machine Learning*, pages 11458–11468. PMLR, 2020. 2, 8, 16

[56] Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. *Advances in Neural Information Processing Systems*, 33:7793–7804, 2020. 17

[57] Yanqiao Zhu, Weizhi Xu, Jinghao Zhang, Yuanqi Du, Jieyu Zhang, Qiang Liu, Carl Yang, and Shu Wu. A survey on graph structure learning: Progress and opportunities. *arXiv e-prints*, pages arXiv–2103, 2021. 2

[58] Daniel Zügner and Stephan Günnemann. Adversarial attacks on graph neural networks via meta learning. 2019. 8