GRAPH AUGMENTATION LEARNING

A PREPRINT

Shuo Yu
School of Software
Dalian University of Technology
Dalian, Liaoning, China, 116620
shuo.yu@ieee.org

Huafei Huang
School of Software
Dalian University of Technology
Dalian, Liaoning, China, 116620
hhuafei@outlook.com

Minh N. Dao
School of Engineering, IT and Physical Sciences
Federation University Australia
Ballarat, VIC, Australia, 3353
m.dao@federation.edu.au

Feng Xia*
School of Engineering, IT and Physical Sciences
Federation University Australia
Ballarat, VIC, Australia, 3353
f.xia@ieee.org

March 18, 2022

ABSTRACT

Graph Augmentation Learning (GAL) provides outstanding solutions for graph learning in handling incomplete data, noise data, etc. Numerous GAL methods have been proposed for graph-based applications such as social network analysis and traffic flow forecasting. However, the underlying reasons for the effectiveness of these GAL methods are still unclear. As a consequence, how to choose optimal graph augmentation strategy for a certain application scenario is still in black box. There is a lack of systematic, comprehensive, and experimentally validated guideline of GAL for scholars. Therefore, in this survey, we in-depth review GAL techniques from macro (graph), meso (subgraph), and micro (node/edge) levels. We further detailedly illustrate how GAL enhance the data quality and the model performance. The aggregation mechanism of augmentation strategies and graph learning models are also discussed by different application scenarios, i.e., data-specific, model-specific, and hybrid scenarios. To better show the outperformance of GAL, we experimentally validate the effectiveness and adaptability of different GAL strategies in different downstream tasks. Finally, we share our insights on several open issues of GAL, including heterogeneity, spatio-temporal dynamics, scalability, and generalization.

Keywords Graph augmentation learning · Graph representation learning · Graph neural networks

1 Introduction

Graph structured data are everywhere. In real world, analyzing graph data has numerous applications and significantly improves human daily life Xia et al. [2021], Yu et al. [2021]. Car-hailing applications analyze traffic network data with graph-based methods to better predict traffic flow Tao et al. [2017]. Social media platforms provide personalized recommendation content to different users Jiang and Chen [2016], Zhang et al. [2019]. Graph-based approaches are designed to optimize the efficiency of resource scheduling in the Internet of Things (IoT) Jia et al. [2018]. Academic networks are analyzed to better enhance the collaboration among scientists Kong et al. [2019], Liu et al. [2020a]. However, real-world graph data are always incomplete due to many reasons such as privacy policies and loss in data collection. Many graph learning methods lack the ability in handling this, thus these methods are generally proposed and verified based on the assumption that data are of high quality. Such neglect leads to suboptimal or even wrong results. Some graph learning methods are designed for special application scenarios that have to consider the data quality in

*Corresponding author.
This work was accepted in The First International Workshop on Graph Learning in IW3C2.

front, such as advisor-advisee recognition Liu et al. [2021a] and fake review detection Cao et al. [2021]. However, these customized graph learning methods can rarely be applied in other tasks. Moreover, redesigning algorithms for another task or application is generally time-consuming, high cost, and there is no guarantee that these limitations can be overcome.

Graph augmentation approaches have been demonstrated to be effective in addressing restrictions in graph learning, e.g., distribution matching is employed to handle graphs with missing attribute Chen et al. [2022]. In fact, augmentation strategies are applied in graph-based tasks for some time Feng et al. [2021a], Lu et al. [2021a], Papp et al. [2021]. Chen *et al.* Chen et al. [2020a] use optimized topology to alleviate the over-smoothing problem in graph representation learning. Graph Augmentation Learning (GAL) is a sort of graph learning approach that integrates a class of augmentation strategies, mechanisms, and models, aiming at overcoming the limitations and promoting the performance of graph learning models. Generally, GAL is designed to enhance the robustness of graph learning models with low-quality data by processing original data or modifying the graph learning models, or both. GAL techniques can improve graph learning methods at all levels of range (i.e., nodes, edges, and graphs), thus achieving big progress in different graph learning applications.

Despite the extensive applications of GAL, the question of what scenario GAL techniques should be employed, and what GAL techniques are applicable, needs to be explored in depth. Unfortunately, there is no systematic, comprehensive, and experimentally validated guideline survey to refer to. Liu *et al.* Liu et al. [2021b] and Xie *et al.* Xie et al. [2021] summarize the usage of graph self-supervised methods, which is one type of GAL technique, to enhance the performance of graph representation learning. Graph self-supervised method can provide a solution for few labels problem. But it is not the only choice in GAL. Survey papers of non-graph structured data in other domains (such as natural language processing) have provided a systematical view of using data augmentation to increase data diversity Feng et al. [2021b], Wen et al. [2021], but cases in graph representation learning are still missing.

To better understand the advantages and features of graph augmentation learning, we review the graph augmentation learning approach from two perspectives, i.e., GAL strategies and GAL application scenarios. Firstly, we analyze the main GAL strategies from the perspective of graph hierarchy, i.e., micro (node/edge), meso (subgraph), and macro (graph) levels. Then, we elaborate on how GAL works in different application scenarios. Specifically, we discuss data-specific, model-specific, and hybrid scenarios. In addition, we also implement experiments to show the outperformance of GAL methods in different application scenarios. A github repository of GAL source codes is also built [2]. Detailedly, our contributions are:

- **Systematical Augmentation Strategy Introduction:** We systematically classify the strategies of existing GAL techniques from three structural levels (node/edge level, subgraph level, graph level) and further illustrate the differences between them through different hierarchies. To the best of our knowledge, this is the first review devoted to GAL.
- **Comprehensive GAL Application Description:** We give a comprehensive description of the existing application scenarios of GAL. In general, the different application scenarios can be considered as the three categories we have divided into, that are data-specific, model-specific, and hybrid scenarios.
- **Verifiable GAL Experiments Guidelines:** We perform experiments on GAL in various application scenarios and verify the effectiveness of GAL by comparing the results. We also analyze the experimental results and give further guidance for scholars to choose optimal GAL strategies.

The rest of this survey is organized as follows. In Section 2, we give the definition of graph augmentation learning. In Section 3, we present the strategies used by GAL from three hierarchy levels, respectively. In Section 4, we discuss how GAL methods perform in different graph learning scenarios. We run GAL experiments and give guidelines in Section 5. In Section 6, we illustrate four open problems of GAL. We conclude the survey in Section 7.

2 What is Graph Augmentation Learning?

As we have illustrated above, GAL can improve graph representation learning results and enhance robustness. Take Graph Convolutional Network (GCN) Kipf and Welling [2017] as an example, the process of GAL is defined as Definition 1.

Definition 1. (*Graph Augmentation Learning):*

[2]https://github.com/yushuowiki/awesome-GAL

For a given graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{X}\}$, where $\mathcal{V}$ is a set containing $|\mathcal{V}|$ nodes, $\mathcal{E}$ is the set of edges showing the links between nodes, and $\mathbf{X}$ is the attribute matrix. GAL technique $\mathrm{Aug}(\cdot)$ aims to learn a mapping function $\Phi : \Phi(\mathcal{V}, \mathcal{E}, \mathbf{X}) \mapsto \mathbb{R}^{|\mathcal{V}| \times d}$ that projects graph nodes to d dimension latent representation $\mathbf{Z}$, where $\Phi = \mathrm{Aug}(\Phi')$, Φ' is the GCN mapping function.

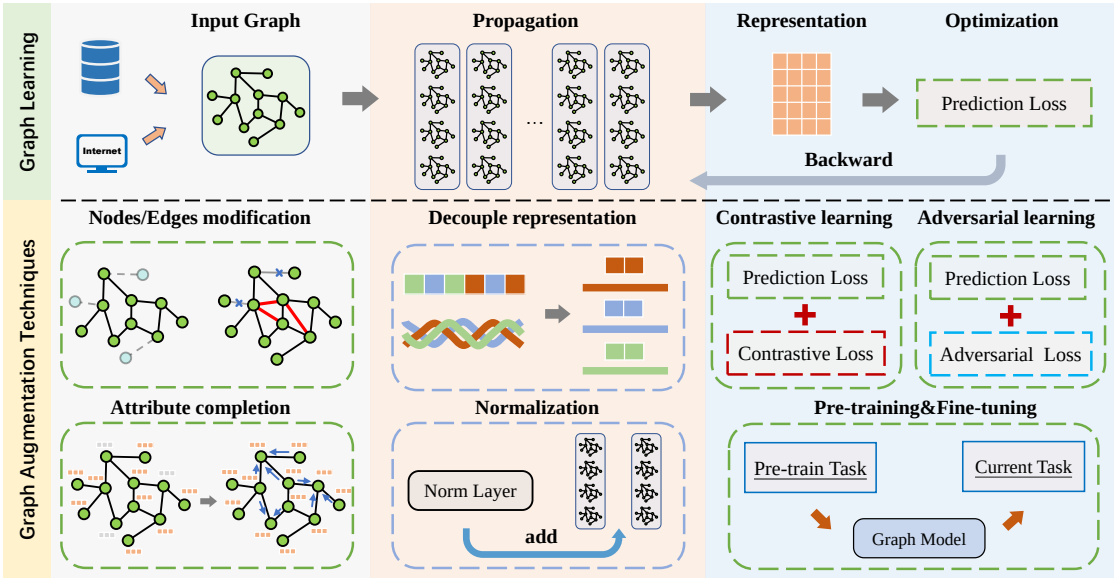


Figure 1: An overall framework for GAL.

3 An Overview of GAL Strategies

In this section, we introduce three kinds of commonly-used augmentation strategies. Augmentation strategies for graph learning can be classified from three categories, i.e., **micro** (node/edge) level, **meso** (subgraph) level, and **macro** (graph) level, which is shown in Figure 2. The differences and characteristics of these augmentation approaches are respectively illustrated.

3.1 Micro-level Augmentation Strategies

Micro-level GAL techniques are dedicated to improving the performance of graph learning by focusing on changing subtleties of graphs (e.g., nodes, edges, and attributes). Specifically, DropGNN Papp et al. [2021] improves the performance of Graph Neural Networks (GNNs) by running multiple experiments independently while dropping nodes randomly. DropEdge Rong et al. [2020] randomly removes graph edges in message passing mechanism to alleviate over-smoothing. In addition, Dropout Srivastava et al. [2014] prevents overfitting through discarding a fixed portion of attributes (or neurons), which has become a widely used method in the field of deep learning. Micro-level GAL techniques have been widely used because the augmentation mechanism is straightforward and easy to realize. Compared to meso and macro level augmentation strategies, micro-level ones are generally in lower computational costs. Meanwhile, implementing micro-level strategies can also achieve satisfying performance. Since such kind of augmentation strategy function on micro-level is not considering graph structures, it can be employed in abundant graphs such as hypergraphs and heterogenous graphs. Accordingly, it has broader applications than the other two kinds, and can effectively improve the robustness as well as generalization ability of models.

3.2 Meso-level Augmentation Strategies

Meso-level GAL techniques, compared with the micro-level one, generally employ path or subgraph information (of target nodes/edges) to enhance graph representation learning. There are some studies using efficient subgraph sampling and computing methods to accelerate meso-level augmentation strategies, thus reducing computational and spatial costs to be affordable. Some studies sample neighbors for each node in the graph, generate low-order subgraph structures, and then execute message passing operations in these subgraphs Zhang and Li [2021], You et al. [2021a]. Similarly, Yu *et al.* Yu et al. [2020] and Xu *et al.* Xu et al. [2020] use motifs to effectively capture the higher-order relationship patterns in complex networks. GraphCrop Wang et al. [2020a], by pruning the subgraph, removes the massive noise in

graphs and achieves better graph classification performance. Compared with micro-level, meso-level GAL techniques pay attention to the wider range of information around nodes in the graph. Subgraph information differs significantly in different kinds of networks. The semantic information of triangle structure is obviously different in social network and traffic network. Therefore, meso-level strategies can extract personalized information in a certain kind of network compared to micro-level ones. On the other hand, the universality is weaker than that of micro-level strategies. As a consequence, this kind of strategy is more popular in recommendation systems and neural language processing.

3.3 Macro-level Augmentation Strategies

Macro-level GAL strategies target to improve graph learning methods from a global view. Concretely, macro-level strategy focuses on graph-level modification or other global operations in graph learning. Wang *et al.* Wang et al. [2020b] use node attribute to construct a K-Nearest Neighbor (KNN) graph, thereby enhancing the model's learning ability without modifying the original graph topology in the graph learning process. At the same time, some studies use joint optimization of multiple highly related tasks to estimate the missing information in a single task Jin et al. [2021a, 2020], others use pre-training Lu et al. [2021a] to accelerate the learning process. In addition, self-supervised methods Sun et al. [2020] can be used as another competitive macro-level augmentation strategy.

Moreover, multiple strategies can also be employed together to achieve better performance Chen et al. [2022], You et al. [2020a], because they can integrate different levels of useful information. The common advantages of GAL strategies are summarized as follows.

- **Auxiliary:** GAL strategies are not newly proposed graph learning methods. Instead, augmentation learning aims to enhance the performance and robustness of graph learning methods.
- **Pluggability:** GAL strategies rarely modify the original graph learning model architecture and meanwhile achieve meaningful improvements. For a given augmented solution, when removing the augmentation part, the model will also function normally. The ability of the original model to complete downstream tasks will not be totally damaged.
- **Generality:** The pluggability of GAL strategies determines generality to some extent. Similar graph learning models (such as GCN and GAT) can share the same GAL strategies. As a result, the improvements in augmentation are also similar.

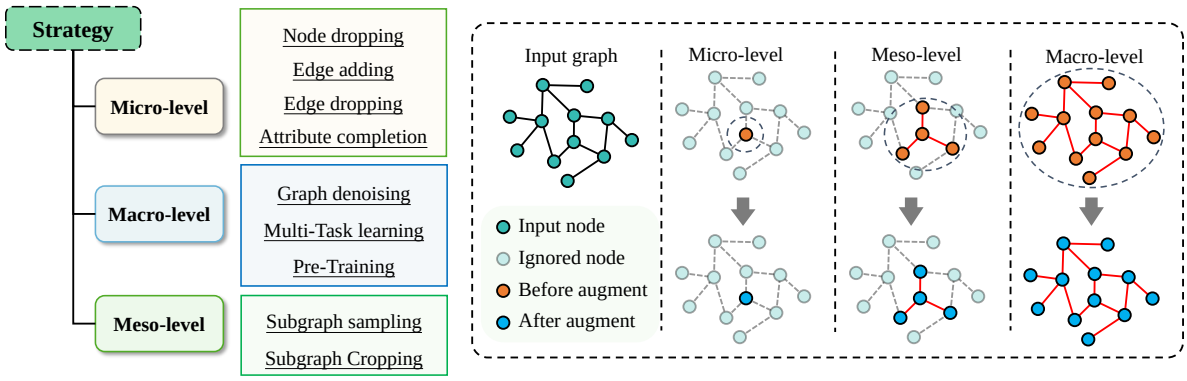


Figure 2: The overview of augmentation strategies in GAL.

4 Augmentation in Graph Learning

In this section, we discuss how the GAL outperforms original graph learning methods. Specifically, three different situations are respectively analyzed: (1) GAL for low-quality data, (2) GAL for model limitation, and (3) GAL for both.

4.1 GAL for Low-quality Data

Real world data are generally low quality because of various reasons such as data collection loss and privacy protection policies. These data limits the performance of graph learning at the first beginning and generally leads to poor learning results. The majority of existing graph learning algorithms presume that the data is almost anomaly-free and of high quality. However, lost and erroneous overlays are unavoidable during data collection and transmission. Moreover,

Table 1: GAL for few-labeled data.

	Types	Methods
Supervised	Pretrain	Hu et al. [2020a], Hu et al. [2020b]
	Resist	Gao et al. [2021], Wang et al. [2020c], Zheng et al. [2020]
Unsupervised	Predefine	You et al. [2020b], Zeng and Xie [2021], Kong et al. [2020]
	Adaptive	Zhu et al. [2021], You et al. [2021b], Suresh et al. [2021]

the graph data naturally contains different degrees of erroneous or noisy information. Once the impact of low-quality data on graph learning tasks is ignored, the optimization process of learning algorithms can easily be hindered or even misled, leading to sub-optimal or even worse results in complex real-world scenarios.

Therefore, GAL techniques need to handle low-quality data while maintaining efficiency and robustness. At present, there have been plenty of studies in Computer Vision (CV) domain to investigate how to overcome low-quality data limitations. But graph data are very different from images, thus few augmentation methods can be employed. Herein, we discuss GAL methods under three main types of data limitations: (1) low-quality node/edge attributes, (2) low-quality graph structures, and (3) few-labeled data.

4.1.1 Low-quality Node/edge Attributes.

Attribute is one of the most important factors in graph learning. Attributes can directly reflect the basic information as well as key features of nodes/edges in the graph. Intuitively, the attribute information in the graph data from the web is sometimes partially corrupted or completely damaged, leading to data low-quality problems. Low-quality attributes often lead to bad performance in almost all kinds of graph learning methods. In order to mitigate the negative impact of low-quality data, Chen *et al.* Chen et al. [2022] develop a novel GNN using a distribution matching technique for attribute-missing graphs. Experimental results indicate the effectiveness of the proposed method in link prediction and attribute completion tasks. HGNN-AC Jin et al. [2021a] use graph attention mechanism to complete the missing attributes of nodes in heterogeneous graphs, avoiding using previous hand-crafted ways to solve this problem.

4.1.2 Low-quality Graph Structures.

A slight change in graph structure can cause significant differences, due to the fact that the data samples are relational to each other. Compared to node attributes, low-quality graph structures pose greater challenges for graph learning. This is because generally connected nodes are regarded to have similar attributes, but not vice versa. Nodes with similar attributes might connect with each other in the further but this is not absolute. Therefore, structural information are usually employed to handle low-quality attribute problem, but attributes may be not that useful in dealing with low-quality structures. It is necessary to denoise the graph structure to extract useful and helpful information correctly.

Nevertheless, modifying or even reconstructing the topology becomes another competitive choice for GAL methods to the graph tasks. Luo *et al.* Luo et al. [2021] introduce a denoising network learning to drop edges that are irrelevant to the downstream tasks, thus improving the robustness and generalization capability in GNNs. Zhao *et al.* Zhao et al. [2021] apply an edge predictor to modify input graphs for better node classification performance. Wang *et al.* Wang et al. [2020a] obtain various subgraphs by cropping operation and generate information-rich augmented graphs to expand the training set. In short, by adding and deleting edges, the structural noise of graph data can be eliminated to a proper extent. In this way, graph learning models can achieve better performance with high robustness at the same time.

4.1.3 Few-labeled Data.

Though a huge amount of data is easily available in the era of big data, sometimes there is only a small percentage is usable. Since the data labeling cost is too high, labeled datasets are often in limited size. Especially in some special research areas, such as biochemistry and medicine, there is only a small amount of available labeled data. The scarcity of labeled data makes graph learning models inevitable prone to overfitting and poor robustness during the training process. Many approaches have been proposed to overcome the limitations caused by insufficient data, i.e., few-labeled data. In the following, we describe how to use GAL to deal with insufficient data from both supervised and unsupervised perspectives. Specifically, we summarize the existing methods in Table 1.

Supervised Methods. In supervised learning settings, when the data amount in the training set is not enough, graph learning methods cannot learn sufficiently in the training process. Moreover, models tend to overfit, leading to poor performance on the test dataset. For few-labeled data, expanding the dataset and enhancing the quality of available data are popular choices. GAL can be used to augment graph data, thus solving the defects caused by data sparsity.

"Pre-training plus fine-tuning" is an excellent paradigm in GAL to solve few-labeled data problems. In this process, the model is first trained on abundant data from other relevant domains and then fine-tuned in the target downstream task using only a small amount of data. Hu *et al.* Hu et al. [2020a] combine node-level and graph-level pre-training and achieve success in molecular property prediction and protein function prediction. GPT-GNN Hu et al. [2020b] employs self-supervised graph generation to pre-train GNN and applies it in recommender systems. In addition, improving the ability to resist overfitting in model training has become another popular solution choice. Injecting perturbations and noise into the graph, such as adding edges and removing nodes, can increase the diversity of the training data and thus contribute to improving the robustness of the model. TADropEdge Gao et al. [2021] drops graph edges adaptively based on their weights by graph spectrum-based computation, and improves generalization performance. Wang *et al.* Wang et al. [2020c] propose three GAL techniques on the graph data by adjusting both the node attributes and the graph structure, thus yielding significant gains for GCN models. There are also exist methods that utilize adversarial training to accomplish data expansion and can be used in large-scale datasets Kong et al. [2020].

Unsupervised methods. Among the unsupervised learning methods, contrast learning has been successfully prevalent in CV, Natural Language Processing (NLP), and other fields. Graph Contrast Learning (GCL), one of the important GAL techniques, has also received increasing attention. The major idea in GCL is to achieve augmentation by contrasting representations extracted from the same graph in different views. Predefined augmentation strategies are popular in GCL and can represent solutions that are designed in advance. It covers different hierarchical levels (nodes, subgraphs, and graphs) and often uses multiple strategies together, such as falling edges, feature masks, subgraph sampling, and graph diffusion You et al. [2020b], Zeng and Xie [2021]. However, how to choose and design an appropriate GAL becomes a question worth discussing. Moreover, the predefined GAL strategies used in most existing solutions lack flexibility. Therefore, dynamic learning of optimal augmentation strategies during the training process is another hot research issue Suresh et al. [2021]. Therefore, adaptive GAL methods Zhu et al. [2021], You et al. [2021b], e.g., automatically selecting the optimal combination of augmentation strategies for the currently running task, are becoming more and more popular.

4.2 GAL for Model Limitation

GNNs are successful paradigms of deep learning on graph data, with excellent performance in social networks, traffic networks, etc. GNNs have become prevalent approaches in graph learning. Despite their excellent learning capability, the limitations of GNN models have been emerging. For example, as the layer number increases, the performance of many powerful GNN models does not get better or even degrades in node classification tasks due to over-smoothing. In addition, the message passing mechanism in GNNs borrows from the 1-dimensional Weisfeiler-Lehman (1-WL) algorithm Leman and Weisfeiler [1968], which absolutely gives GNNs excellent capabilities, but also makes them unable to distinguish some special isomorphic graphs. Therefore, GNNs expressive power is accordingly limited in many situations. In this section, we discuss how to solve the above problems through GAL methods.

4.2.1 Over-smoothing

As the depth of the model increases, the representations of all nodes in the graph gradually fused and eventually become indistinguishable with iterative message passing. This is the so-called over-smoothing phenomenon, and such a phenomenon generally leads to failure in the task Li et al. [2018]. To cope with over-smoothing, adding noise to the data or giving regularization terms to the optimization process are effective means.

Table 2: GAL for over-smoothing.

	Types	Methods
Locality	Perturbation	Liu et al. [2021c], Lu et al. [2021b]
	Propagation	Chen et al. [2020b], Yang et al. [2021], Hu et al. [2020c]
	Transformation	Liu et al. [2020b], Min et al. [2020]
	Subgraph-based	Zeng et al. [2020], Zhou et al. [2020]
Globality	Regularization	Zheng et al. [2021], Chen et al. [2020a], Velickovic et al. [2019]

Locality. From a local perspective, we classify the GAL methods into perturbation, propagation, transformation, and subgraph-based methods. To facilitate understanding, we show how propagation and transformation work in graph convolution layer by follows:

$$
\begin{aligned}
a_i^{(l)} &= \textbf{Propagation}\left(\left\{h_i^{(l-1)}\right\} \cup \left\{h_j^{(l-1)} \mid j \in \mathcal{N}_i\right\}\right) \\
h_i^{(l)} &= \textbf{Transformation}\left(h_i^{(l)}\right),
\end{aligned} \tag{1}
$$

where $h_i^{(l)}$ represents the feature of node i in l-th layer, and $\mathcal{N}_i$ is neighbor node set from node i. Next we introduce these types of above-mentioned methods.

Perturbation, as the name suggests, this sort of method randomly changes the nodes/edges contained in the original graph during the training process, or masks some of the nodes from computation. For instance, Liu *et al.* Liu et al. [2021c] dynamically add and remove edges based on the graph motif structure to preserve the graph entropy as much as possible. SkipNode Lu et al. [2021b], on the other hand, lets some nodes skip the graph convolution operation in a certain probability, preserving the features for the next use, thus mitigating the gradient vanishing and weight over-decaying issues. **Propagation** is an important component of convolution on a graph, i.e., the way nodes aggregate neighbors, typically includes direct neighbors or multi-hop neighbors. Some special propagation methods are carefully designed to prevent over-smoothing. GCNII Chen et al. [2020b] aggregates the initial and the previous layer representation to the current layer by adding a residual connection in training. Yang *et al.* Yang et al. [2021] then propose graph conjugate convolution based on GCNII. **Transformation** represents the way how features are transformed in graph convolution and plays a key role in graph learning. Some methods try to solve over-smoothing by improving the graph convolution computation formula. Liu *et al.* Liu et al. [2020b] successfully increase the model depth by decoupling graph convolution while maintaining performance and avoiding over-smoothing. Scattering GCN Min et al. [2020] integrates geometric scattering transforms and residual convolutions to augment conventional GCNs. Subgraph-based method is a type of approach that improves the graph representation learning capability by considering **subgraph information**. For example, SHADOW-GNN Zeng et al. [2020] samples subgraphs centered on each node in the whole graph and then builds an L-layer GNN operating on subgraph instead of the whole graph. Zhou *et al.* Zhou et al. [2020] introduce a differentiable group normalization to GNNs, which not only normalizes nodes within the sample group independently but also separates node distributions among different groups to alleviate over-smoothing.

Globality. Regularization, which is widely used in machine learning to prevent overfitting, can also mitigate the over-smoothing problem from a global perspective. By adding a regularization term to the loss function in a deep model, the node representations' fast convergence can be prevented. For example, TGCL Zheng et al. [2021] adds a graph contrastive layer guided by topology before the output layer of the GNN model, reducing the negative impact of remote nodes with similar topology features. In addition, Chen *et al.* Chen et al. [2020a] point out that over-smoothing is caused by an excessive mixture of information and noise, and use a regularization term to increase the received information and decrease the received noise. There are also some studies using other types of GAL strategies. The cause of over-smoothing is regarded as the entanglement of propagation and transformation Liu et al. [2020b]. By decoupling these two operations, the model can avoid over-smoothing. In general, most of these approaches mentioned above can be unified as follows:

$$
\begin{aligned}
\mathbf{H}^{l+1} &= \sigma\left(\mathcal{T}_1(\mathbf{A})\mathcal{T}_2(\mathbf{H}^l)\mathbf{W}^l\right) \\
\mathcal{L}_{total} &= \mathcal{L}_{rep} + \mathcal{L}_{reg},
\end{aligned} \tag{2}
$$

where $\mathbf{A}$ refers to the adjacency matrix, $\mathbf{H}^l$ and $\mathbf{W}^l$ are the weight and hidden representation of layer l, respectively. $\mathcal{T}_1, \mathcal{T}_2 \in \mathcal{T}$ represent specifical augmentation strategies like drop node, drop edge or mask feature, etc., $\mathcal{L}_{rep}$ denote the representation learning loss, and $\mathcal{L}_{reg}$ is a regularization term to relieve over-smoothing.

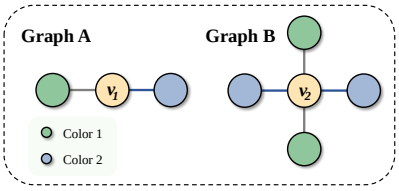
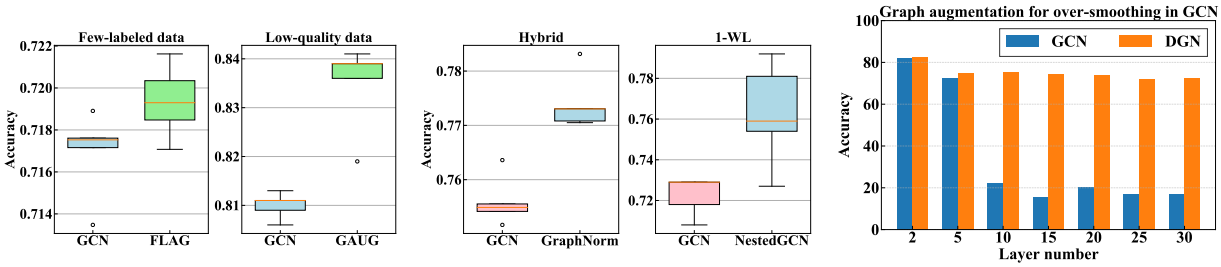


Figure 3: Two networks that traditional GNNs fail to distinguish.

4.2.2 1-WL

GNN models have limitations in representation ability. Many pieces of research show that GNNs are neural variants of 1-WL, the algorithm used to test whether two graphs are isomorphic. They also show that GNNs do not exceed the expressive power of 1-WL Morris et al. [2019], Xu et al. [2018]. For instance, WL-based GNNs, such as GraphSAGE Hamilton et al. [2017], JKNet Duvenaud et al. [2015], and GAT Velickovic et al. [2018], cannot distinguish



(a) Node classification (b) Graph classification (c) Over-smoothing

Figure 4: Graph augmentation learning results in different tasks.

the two graphs in Figure 3. But the difference can be identified by 1-WL. In this section, we discuss how to improve models through GAL and obtain GNNs with comparable or even better capabilities than 1-WL.

Some GNN algorithms fail to distinguish the two graphs because they use non-injective readout functions (e.g., mean, max, or sum) to transform node information to graph representation, and thus they cannot adapt to complex situations. GAL can be an outstanding choice to tackle this challenge. The approaches of such GALs vary, but most of them have one thing in common, i.e., they improve the way the aggregation and transformation operation in GNNs. In the following, we present how to address the limitations of GNNs' representation ability through GAL techniques.

One feasible attempt is to design an excellent message passing scheme to obtain a GNN model with better performance and to catch up (or outperform) 1-WL in distinguishing non-isomorphic graphs. You *et al.* You et al. [2021a] propose ID-GNN, which is more powerful than 1-WL test. ID-GNN assigns a unique identity to each node, which is used to distinguish the node from other nodes in its neighbors, and the node's identity is taken into account to update the node's embedding during the GNN's message passing process. Maron *et al.* Maron et al. [2019] design a simple GNN model that utilizes a multilayer perceptron (MLP) and matrix multiplication, which has a provable 3-WL expressive power while maintaining scalability.

Considering that the GNN model is a variant of 1-WL, one way to improve it is to design a GNN based on k ($k > 1$) dimensional Weisfeiler-Lehman (k-WL), which can be implemented by subgraph-based or local structural augmentation technology. Some studies claim that the reason behind such limitation of GNNs lies in the inability to adequately capture graph topology information. It can be resolved by subgraph sampling and other GAL strategies. These studies usually introduce graph structure information during feature aggregation iteration by comparing WL algorithms and GNNs through theoretical analysis. In particular, Zhang and Li Zhang and Li [2021] propose NestedGNN, which embeds nodes by encoding a rooted subgraph instead of a rooted subtree. NestedGNN is more powerful than message passing GNNs and 1-WL. Inspired by k-WL, k-GNN Morris et al. [2019] considers multi-scale higher-order graph structures and strictly outperforms GNNs in the ability to distinguish non-isomorphic graphs/subgraphs.

4.3 GAL for Hybrid Scenario

In most cases, low-quality data and model limitation both exist in graph learning tasks. But most current studies are devoted to solving the problem from one perspective. Those studies directed at solving the graph learning model problem are specified in terms of expanding node perceptual field, improving expressiveness, and preventing over-smoothing. However, these works generally consider graph data are perfect, which is impracticable in real scenarios, like bot users and invalid users in social networks. Moreover, there may be insufficient supervised labels in the data, such as drug network data and biomolecular network data. Therefore, in real-world scenarios, the challenges of data and models appear in a mixed manner.

Designing solutions for only one class of problems while ignoring the other one is clearly lacking in generalization, and will eventually lead to suboptimal results. Thus, we need solutions that can solve the model problem and at the same time can accommodate the shortcomings of low-quality graph data. Numerous hybrid scenarios require augmentation techniques, but only a few existing works study such scenarios simultaneously.

In this regard, Jin *et al.* Jin et al. [2021b] analyze the limitations of Laplace operator in GCN and point out that the operator is not robust to the noise links existing in graph data. Therefore, they design a new operator and integrate it into the existing GCN backbone, to improve the noise tolerance of data and the performance of the model. Normalization technology can accelerate the optimization process and has been widely used in deep learning, especially in the fields

Table 3: Statistic information of datasets used in the experiments.

Datasets	#Graphs	#nodes	#edges
ogbn-Arxiv	1	169,343	1,166,243
Cora	1	2,708	5,429
Datasets	**#Graphs**	**Avg. #nodes**	**Avg. #edges**
ogbg-MOLHIV	41,127	25.5	27.5
MUTAG	188	17.93	19.79

of CV and NLP. For graph data, GraphNorm Cai et al. [2021] successfully adapts to noise and high variance through processing graph batch in tasks for complex data environments. The generalization ability of the model is also enhanced.

5 How to Understand GAL?

In this section, we further verify the performance of GAL techniques and then give sound advice on choosing GAL strategies. We have presented the strategies of GAL from three perspectives in Section 3 and illustrated that GAL can solve different problems for graph learning in Section 4. It is obvious that GAL, as an efficient tool, can remedy the shortcomings of existing graph learning methods in various scenarios. However, there still remain many questions. How to choose GAL strategies in certain applications? How much degree GAL can improve the performance?

Since GAL can overcome the limitations existing in the original graph learning method, More explicitly, GAL can bring a direct experimental performance gain for the task. Therefore, we run experiments with the graph learning approach in each of the scenarios mentioned in this paper and quantitatively evaluate the gains brought by the GAL to the original methods. Afterward, we analyze the experiment results and give guiding suggestions for future researchers on graph augmentation techniques in various scenarios.

For this purpose, we select datasets for performing GAL experiments for mentioned scenarios in Section 4. These challenging cases include few-labeled data, low-quality data, over-smoothing, 1-WL limitation, and hybrid scenario, respectively. The dataset used in the experiments are Cora Kipf and Welling [2017], ogbn-Arxiv Hu et al. [2020d], ogbg-MOLHIV Hu et al. [2020d], and MUTAG Morris et al. [2020] as shown in Table 3.

In all the experiments, we choose GCN as the original solution, which is widely used in the study of graph learning. After augmentation, we get augmented GCN methods and then use them to run experiments. The results are compared with those output by vanilla GCN. Firstly, in node classification (NC) task (with few-labeled data) we choose the method proposed by Kong *et al.* Kong et al. [2020]. For the limitation of low-quality data, GAUG Zhao et al. [2021] is implemented. In the graph classification (GC) task, we use NestedGCN Zhang and Li [2021] to augment the GCN in order to cope with the 1-WL capability limitation. For the hybrid case, we apply GraphNorm proposed by Cai *et al.* Cai et al. [2021]. In addition, for the above four methods, we run them five times using different random seeds. Finally, we use DGN Zhou et al. [2020] to handle the over-smoothing problem in node representation.

Figure 4 shows the results of experiments. FLAG [3] augments the features of graph nodes by introducing adversarial perturbations in training. It improves the GNN performance while preventing overfitting due to few-labeled data. For the low-quality data problem, GAUG [4] proposes an edge predictor that achieves performance improvement by denoising graph edges, i.e., removing noisy edges and adding edges that potentially missing from the original graph. GraphNorm [5] proposes a learnable shift to normalize the node representation in the graph, handles the noise in the graph that cannot be coped with BatchNorm. Such an improvement also avoids the expressiveness degradation caused by the shift operation brought in InstanceNorm and allows the algorithm to converge faster. NestedGCN [6] first extracts subgraphs for each node, then learns the subgraph representation, and finally, through a message passing mechanism, gets the graph representation. Such an approach makes it beyond the 1-WL power. DGN [7] takes into account the distribution of different communities in the graph and takes different normalization operations for various communities, thus resist to over-smoothing. Overall, we can see that the task metrics of the backbone are improved after using GAL techniques, which reflects the efficiency of graph augmentation learning.

[3] https://github.com/devnkong/FLAG
[4] https://github.com/zhao-tong/GAug
[5] https://github.com/lsj2408/GraphNorm
[6] https://github.com/muhanzhang/NestedGNN
[7] https://github.com/Kaixiong-Zhou/DGN

The above experimental results reflect that GAL can enhance GNN in different tasks. Here we give guiding suggestions for using GAL technology based on the experimental results and related work.

- **Few-labeled data.** Training techniques can be employed to compensate for the lack of labeled data, e.g., pre-training plus fine-tuning or contrastive learning. It is also possible to make the model resistant to overfitting by improving its robustness. In addition, adversarial training is a potential direction.
- **Low-quality data.** Complementing attribute techniques can be used to address the challenge of missing information in graphs. Corrupted graph topology can be repaired by denoising, thus performing well in downstream tasks.
- **Over-smoothing.** For the over-smoothing problem, it is suggested to add a regularization term to the model, introduce subgraph information, or use decoupled representation learning.
- **1-WL.** It is recommended that considering the introduction of identity information or using higher-order structure feature around nodes, which can go beyond the 1-WL representation limitation.

6 Open Issues

Despite existing studies that have made progress in GAL, there still remain many open issues in terms of heterogeneity, spatio-temporal network, scalability, generalization.

6.1 Heterogeneity

Most graph augmentation learning methods focus on homogeneous graphs including augmenting nodes, structural attributes, or models. However, the continuous development of graph learning methods is still difficult to handle the problem of heterogeneity in graph data. Heterogeneous graphs contain a lot more information than homogeneous graphs. Most graphs are heterogeneous such as knowledge graphs and citation networks, so there is of great significance to handle heterogeneity. At present, there is a lack of studies about GAL on heterogeneous graphs. Due to the diversity of types of nodes and edges in heterogeneous graphs, it will be more difficult to study GAL than that on homogeneous graphs. The complexity problems brought by various attributes and the design of augmenting model should be paid more attention to.

6.2 Spatio-temporal Dynamics

Spatio-temporal network is a common graph data type in daily life, such as traffic networks. Data loss or data inaccuracy frequently occur in the process of data collection. The data loss and inaccuracy at one certain single time point will have a negative impact on the downstream task results. Spatio-temporal networks vary over time, the loss and inaccuracy happen every time. As a consequence, the increasing complexity of the data has a significant influence on GAL techniques. Moreover, spatio-temporal networks contain two aspects of time and space. How to balance the augmentation strategies of the two aspects and meanwhile design an effective model remain an open issue.

6.3 Scalability

Large-scale networks are ubiquitous in the real world, consisting of billions of nodes. As a result, graph learning models are generally time-consuming and have unaffordable space complexity. The corresponding problem definitely also exists in GAL, or even worse. In large-scale graphs, as the number of nodes or edges increases, the scale of augmentation data will also increase. But till now, there is no effective parallel solution for GAL. The problem of high cost, the selection of augmentation strategy, and the optimization of augmentation model are the main problems when facing scalability in GAL.

6.4 Generalization

Although GAL approaches have been proved to be effective in many different tasks, most approaches or strategies are designed for specific tasks or datasets. As it is known, graphs are very different in many domains, no matter in scale or structure. The consequence of this is that the generalization of the model is reduced. In GAL, the type and method of augmenting data will affect the effect of model generalization. If datasets and tasks are analyzed ahead of time, more overhead can be incurred. The generalization of the model still lacks theoretical exploration, and the stability of the algorithm also plays an important role in generalization.

7 Conclusion

Graph Augmentation Learning (GAL) techniques have received a lot of attention for their outperformance in different tasks. This survey provides a comprehensive overview of GAL techniques. Augmentation strategies are divided into macro, meso, and micro levels, respectively. The survey also summarizes GAL's ability in enhancing the robustness of graph learning models. Specifically, the outperformance of GAL is discussed. Three different situations namely low-quality data, model limitation, and hybrid are analyzed. In addition, this survey demonstrates the positive effects of GAL technologies through comparative experiments. Moreover, guidelines of suitable and feasible GAL strategies for different tasks are also provided. To the best of our knowledge, this survey paper is the first to discuss GAL techniques from such a systematical, comprehensive, and verifiable view.

GAL is a flourishing research field, which still remains many open issues as we have discussed in this survey paper, such as heterogeneity, dynamics, scalability, generalization, etc. Compared with other Euclidean data, the complexity of graph data grows exponentially faster. Low-quality graph data, such as the missing of relations or nodes within graph data, will have a much worse impact on downstream task accuracy than other Euclidean data. Therefore, the significance of GAL should be highly focused, and more efforts should be contributed to GAL as well.

8 Acknowledgments

This work is partially supported by the National Key Research and Development Program of China under Grant No. 2021ZD0112400, and National Natural Science Foundation of China under Grant No. 62102060. The authors would like to thank Yin Peng for his help with the experiments.

References

Feng Xia, Ke Sun, Shuo Yu, Abdul Aziz, Liangtian Wan, Shirui Pan, and Huan Liu. Graph learning: A survey. *IEEE Transactions on Artificial Intelligence*, 2(2):109–127, 2021.

Shuo Yu, Qing Qing, Chen Zhang, Ahsan Shehzad, Giles Oatley, and Feng Xia. Data-driven decision-making in COVID-19 response: A survey. *IEEE Transactions on Computational Social Systems*, 8(4):1016–1029, 2021.

Jing Tao, Pinghui Wang, Xiaohong Guan, and Wenjun Hu. Al-bitmap: Monitoring network traffic activity graphs on high speed links. *Information Sciences*, 408:162–175, 2017.

Shan Jiang and Hsinchun Chen. NATERGM: A model for examining the role of nodal attributes in dynamic social media networks. *IEEE Transactions on Knowledge and Data Engineering*, 28(3):729–740, 2016.

Dongyu Zhang, Teng Guo, Hanxiao Pan, Jie Hou, Zhitao Feng, Liang Yang, Hongfei Lin, and Feng Xia. Judging a book by its cover: The effect of facial perception on centrality in social networks. In *Proceedings of The Web Conference*, pages 2290–2300, 2019.

Yizhen Jia, Yinhao Xiao, Jiguo Yu, Xiuzhen Cheng, Zhenkai Liang, and Zhiguo Wan. A novel graph-based mechanism for identifying traffic vulnerabilities in smart home iot. In *IEEE Conference on Computer Communications*, pages 1493–1501, 2018.

Xiangjie Kong, Yajie Shi, Shuo Yu, Jiaying Liu, and Feng Xia. Academic social networks: Modeling, analysis, mining and applications. *Journal of Network and Computer Applications*, 132:86–103, 2019.

Jiaying Liu, Jing Ren, Wenqing Zheng, Lianhua Chi, Ivan Lee, and Feng Xia. Web of scholars: A scholar knowledge graph. In *Proceedings of International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2153–2156, 2020a.

Jiaying Liu, Feng Xia, Lei Wang, Bo Xu, Xiangjie Kong, Hanghang Tong, and Irwin King. Shifu2: A network representation learning based model for advisor-advisee relationship mining. *IEEE Transactions on Knowledge and Data Engineering*, 33(4):1763–1777, 2021a.

Chen Cao, Shihao Li, Shuo Yu, and Zhikui Chen. Fake reviewer group detection in online review systems. In *International Conference on Data Mining Workshops*, pages 935–942. IEEE, 2021.

Xu Chen, Siheng Chen, Jiangchao Yao, Huangjie Zheng, Ya Zhang, and Ivor W. Tsang. Learning on attribute-missing graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(2):740–757, 2022.

Fuli Feng, Xiangnan He, Jie Tang, and Tat-Seng Chua. Graph adversarial training: Dynamically regularizing based on graph structure. *IEEE Transactions on Knowledge and Data Engineering*, 33(6):2493–2504, 2021a.

Yuanfu Lu, Xunqiang Jiang, Yuan Fang, and Chuan Shi. Learning to pre-train graph neural networks. In *Proceedings of AAAI Conference on Artificial Intelligence*, pages 4276–4284, 2021a.

Pál András Papp, Karolis Martinkus, Lukas Faber, and Roger Wattenhofer. Dropgnn: Random dropouts increase the expressiveness of graph neural networks. In *Proceedings of Conference on Neural Information Processing Systems*, 2021.

Deli Chen, Yankai Lin, Wei Li, Peng Li, Jie Zhou, and Xu Sun. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *Proceedings of AAAI Conference on Artificial Intelligence*, volume 34, pages 3438–3445, 2020a.

Yixin Liu, Shirui Pan, Ming Jin, Chuan Zhou, Feng Xia, and Philip S. Yu. Graph self-supervised learning: A survey. *CoRR*, abs/2103.00111, 2021b.

Yaochen Xie, Zhao Xu, Zhengyang Wang, and Shuiwang Ji. Self-supervised learning of graph neural networks: A unified review. *CoRR*, abs/2102.10757, 2021.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard H. Hovy. A survey of data augmentation approaches for NLP. In *ACL/IJCNLP*, volume ACL/IJCNLP 2021, pages 968–988, 2021b.

Qingsong Wen, Liang Sun, Fan Yang, Xiaomin Song, Jingkun Gao, Xue Wang, and Huan Xu. Time series data augmentation for deep learning: A survey. In *Proceedings of International Joint Conference on Artificial Intelligence*, pages 4653–4660, 2021.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proceedings of International Conference on Learning Representations*, 2017.

Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. Dropedge: Towards deep graph convolutional networks on node classification. In *Proceedings of International Conference on Learning Representations*, 2020.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Muhan Zhang and Pan Li. Nested graph neural networks. In *Proceedings of Conference on Neural Information Processing Systems*, 2021.

Jiaxuan You, Jonathan M. Gomes-Selman, Rex Ying, and Jure Leskovec. Identity-aware graph neural networks. In *Proceedings of AAAI Conference on Artificial Intelligence*, pages 10737–10745, 2021a.

Shuo Yu, Feng Xia, Jin Xu, Zhikui Chen, and Ivan Lee. OFFER: A motif dimensional framework for network representation learning. In *Proceedings of ACM International Conference on Information and Knowledge Management*, pages 3349–3352. ACM, 2020.

Jin Xu, Shuo Yu, Ke Sun, Jing Ren, Ivan Lee, Shirui Pan, and Feng Xia. Multivariate relations aggregation learning in social networks. In *Proceedings of ACM/IEEE Joint Conference on Digital Libraries*, pages 77–86. ACM, 2020.

Yiwei Wang, Wei Wang, Yuxuan Liang, Yujun Cai, and Bryan Hooi. Graphcrop: Subgraph cropping for graph classification. *CoRR*, abs/2009.10564, 2020a.

Xiao Wang, Meiqi Zhu, Deyu Bo, Peng Cui, Chuan Shi, and Jian Pei. AM-GCN: adaptive multi-channel graph convolutional networks. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1243–1253, 2020b.

Di Jin, Cuiying Huo, Chundong Liang, and Liang Yang. Heterogeneous graph neural network via attribute completion. In *Proceedings of The Web Conference*, pages 391–400, 2021a.

Wei Jin, Yao Ma, Xiaorui Liu, Xianfeng Tang, Suhang Wang, and Jiliang Tang. Graph structure learning for robust graph neural networks. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 66–74, 2020.

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 AAAI Conference on Artificial Intelligence*, volume 34, pages 5892–5899, 2020.

Yuning You, Tianlong Chen, Zhangyang Wang, and Yang Shen. When does self-supervision help graph convolutional networks? In *Proceedings of International Conference on Machine Learning*, volume 119, pages 10871–10880, 2020a.

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 ACM International Conference on Web Search and Data Mining*, pages 779–787, 2021.

Tong Zhao, Yozen Liu, Leonardo Neves, Oliver J. Woodford, Meng Jiang, and Neil Shah. Data augmentation for graph neural networks. In *Proceedings of AAAI Conference on Artificial Intelligence*, pages 11015–11023, 2021.

W Hu, B Liu, J Gomes, M Zitnik, P Liang, V Pande, and J Leskovec. Strategies for pre-training graph neural networks. In *Proceedings of International Conference on Learning Representations*, 2020a.

Ziniu Hu, Yuxiao Dong, Kuansan Wang, Kai-Wei Chang, and Yizhou Sun. GPT-GNN: generative pre-training of graph neural networks. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1857–1867, 2020b.

Zhan Gao, Subhrajit Bhattacharya, Leiming Zhang, Rick S. Blum, Alejandro Ribeiro, and Brian M. Sadler. Training robust graph neural networks with topology adaptive edge dropping. *CoRR*, abs/2106.02892, 2021.

Yiwei Wang, Wei Wang, Yuxuan Liang, Yujun Cai, Juncheng Liu, and Bryan Hooi. Nodeaug: Semi-supervised node classification with data augmentation. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 207–217, 2020c.

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 *Proceedings of International Conference on Machine Learning*, volume 119, pages 11458–11468, 2020.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. In *Proceedings of Conference on Neural Information Processing Systems*, volume 33, pages 5812–5823, 2020b.

Jiaqi Zeng and Pengtao Xie. Contrastive self-supervised learning for graph classification. In *Proceedings of AAAI Conference on Artificial Intelligence*, volume 35, pages 10824–10832, 2021.

Kezhi Kong, Guohao Li, Mucong Ding, Zuxuan Wu, Chen Zhu, Bernard Ghanem, Gavin Taylor, and Tom Goldstein. FLAG: adversarial data augmentation for graph neural networks. *CoRR*, abs/2010.09891, 2020.

Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Graph contrastive learning with adaptive augmentation. In *Proceedings of The Web Conference*, pages 2069–2080, 2021.

Yuning You, Tianlong Chen, Yang Shen, and Zhangyang Wang. Graph contrastive learning automated. In *Proceedings of International Conference on Machine Learning*, volume 139, pages 12121–12132, 2021b.

Susheel Suresh, Pan Li, Cong Hao, and Jennifer Neville. Adversarial graph augmentation to improve graph contrastive learning. In *Proceedings of Conference on Neural Information Processing Systems*, 2021.

AA Leman and Boris Weisfeiler. A reduction of a graph to a canonical form and an algebra arising during this reduction. *Nauchno-Technicheskaya Informatsiya*, 2(9):12–16, 1968.

Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of AAAI Conference on Artificial Intelligence*, pages 3538–3545. AAAI Press, 2018.

Xue Liu, Dan Sun, and Wei Wei. A graph data augmentation strategy with entropy preserving. *CoRR*, abs/2107.06048, 2021c.

Weigang Lu, Yibing Zhan, Ziyu Guan, Liu Liu, Baosheng Yu, Wei Zhao, Yaming Yang, and Dacheng Tao. Skipnode: On alleviating over-smoothing for deep graph convolutional networks. *CoRR*, abs/2112.11628, 2021b.

Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. Simple and deep graph convolutional networks. In *Proceedings of International Conference on Machine Learning*, volume 119, pages 1725–1735, 2020b.

Liang Yang, Chuan Wang, Junhua Gu, Xiaochun Cao, and Bingxin Niu. Why do attributes propagate in graph convolutional neural networks? In *AAAI Conference on Artificial Intelligence*, volume 35, pages 4590–4598, 2021.

Ruiqi Hu, Shirui Pan, Guodong Long, Qinghua Lu, Liming Zhu, and Jing Jiang. Going deep: Graph convolutional ladder-shape networks. In *Proceedings of AAAI Conference on Artificial Intelligence*, volume 34, pages 2838–2845, 2020c.

Meng Liu, Hongyang Gao, and Shuiwang Ji. Towards deeper graph neural networks. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 338–348, 2020b.

Yimeng Min, Frederik Wenkel, and Guy Wolf. Scattering GCN: overcoming oversmoothness in graph convolutional networks. In *Proceedings of Conference on Neural Information Processing Systems*, 2020.

Hanqing Zeng, Muhan Zhang, Yinglong Xia, Ajitesh Srivastava, Rajgopal Kannan, Viktor K. Prasanna, Long Jin, Andrey Malevich, and Ren Chen. Deep graph neural networks with shallow subgraph samplers. *CoRR*, abs/2012.01380, 2020.

Kaixiong Zhou, Xiao Huang, Yuening Li, Daochen Zha, Rui Chen, and Xia Hu. Towards deeper graph neural networks with differentiable group normalization. In *Proceedings of Conference on Neural Information Processing Systems*, pages 4917–4928, 2020.

Lecheng Zheng, Dongqi Fu, and Jingrui He. Tackling oversmoothing of gnns with contrastive learning. *CoRR*, abs/2110.13798, 2021.

Petar Velickovic, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R. Devon Hjelm. Deep graph infomax. In *Proceedings of International Conference on Learning Representations*, 2019.

Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of AAAI Conference on Artificial Intelligence*, volume 33, pages 4602–4609, 2019.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *Proceedings of International Conference on Learning Representations*, 2018.

William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proceedings of Conference on Neural Information Processing Systems*, pages 1024–1034, 2017.

David Duvenaud, Dougal Maclaurin, Jorge Aguilera-Iparraguirre, Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P. Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Proceedings of Conference on Neural Information Processing Systems*, pages 2224–2232, 2015.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *Proceedings of International Conference on Learning Representations*, 2018.

Haggai Maron, Heli Ben-Hamu, Hadar Serviansky, and Yaron Lipman. Provably powerful graph networks. In *Proceedings of Conference on Neural Information Processing Systems*, volume 32, pages 2156–2167, 2019.

Ming Jin, Heng Chang, Wenwu Zhu, and Somayeh Sojoudi. Power up! robust graph convolutional network via graph powering. In *Proceedings of AAAI Conference on Artificial Intelligence*, pages 8004–8012, 2021b.

Tianle Cai, Shengjie Luo, Keyulu Xu, Di He, Tie-Yan Liu, and Liwei Wang. Graphnorm: A principled approach to accelerating graph neural network training. In *Proceedings of International Conference on Machine Learning*, volume 139, pages 1204–1215, 2021.

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. In *Proceedings of Conference on Neural Information Processing Systems*, 2020d.

Christopher Morris, Nils M. Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. Tudataset: A collection of benchmark datasets for learning with graphs. *CoRR*, abs/2007.08663, 2020.