IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 5 March 2025, accepted 26 March 2025, date of publication 1 April 2025, date of current version 9 April 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3555997

RESEARCH ARTICLE

Augmented Graph Convolutional Network for Enhancing Label Reachability

XIANGYI WANG, FENGJUN ZHANG, WEI TENG, AND BAODA LIU
School of Computer, Jiangsu University of Science and Technology, Zhenjiang, Jiangsu 212100, China

Corresponding author: Fengjun Zhang (jszhfj@nuaa.edu.cn)

This work was supported by the Research Start-Up Funding for High-Level Talent of Jiangsu University of Science and Technology under Grant 1132932301 and Grant 1022932208.

ABSTRACT Graph Convolutional Networks (GCNs) have emerged as a leading approach for semi-supervised node classification. However, due to the uneven distribution of labeled nodes in graphs, only a limited subset of unlabeled nodes can directly access the labeled nodes. This disconnection between the labeled and unlabeled nodes significantly hinders GCNs performance. To address this challenge, we propose an **A**ugmented **G**raph **C**onvolutional **N**etwork for enhancing **L**abeled **R**eachability (AGCN-LR), which is designed to improve information flow to low-degree nodes by strengthening their connectivity with labeled nodes. The core of AGCN-LR comprises two parts. First, we selectively connect low-degree nodes to labeled nodes by calculating feature similarity and shortest path distances to generate augmented graphs, thereby enhancing label reachability. Second, we leverage GCNs to learn these augmented graphs and integrate embeddings from these graphs through an attention mechanism to bolster overall performance. Furthermore, to capture consistent information across augmented graphs, we incorporate a tailored contrastive loss function, facilitating consistent contextual learning across different augmented graphs. Experimental results demonstrate that AGCN-LR substantially enhances performance in semi-supervised node classification tasks across various benchmark datasets, exhibiting stronger advantages in sparse label and noisy graph scenarios.

INDEX TERMS Graph convolutional networks, label reachability, contrastive learning, node classification, semi-supervised learning.

I. INTRODUCTION

Graphs are a fundamental data structure that effectively model complex relationships and dependencies among entities in various domains [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18]. They serve as the foundational representation for expressing intricate structures across a wide range of scientific fields. For example, graphs are used to represent user relationships in social networks and gene interactions in bioinformatics [19], [20], [21]. As a result, graph data processing techniques have attracted significant research attention in recent years.

With the development of deep learning, Graph Convolutional Networks (GCNs), leveraging their efficient convolutional operations and message-passing mechanisms [1], [3], [22], have become a dominant approach for semi-supervised node classification tasks [6], [23], [24], [25], [26]. Studies have shown that the success of GCNs lies in their ability to establish connections between labeled and unlabeled nodes through convolution over graph structures, thereby aligning their distributions and enabling more accurate predictions for unlabeled nodes. During training, GCNs optimize by minimizing the supervised loss on labeled nodes, and once trained, the model predicts labels for the unlabeled nodes. Essentially, a K-layer GCN allows labeled nodes to aggregate feature information from their k-hop neighbors, thereby improving the representation quality of unlabeled nodes [1], [3], [6], [23], [27].

However, recent research [5], [6], [24], [27], [28], [29], [30], [31] highlights significant challenges faced by GCNs because of the differences in topological positions between low-degree and high-degree nodes (the uneven distribution). Low-degree nodes are often far from labeled nodes, while high-degree nodes tend to be closer [1], [6], [27], [30],

The associate editor coordinating the review of this manuscript and approving it for publication was Cesar Vargas-Rosales.

[31]. The limited depth of GCN models restricts low-degree nodes from accessing valuable information from distant labeled nodes, leading to suboptimal distribution alignment in shallow GCN architectures [3], [23], [30], [32]. As a result, trained GCN models primarily focus on nodes near labeled ones, with weaker classification performance for nodes that are less accessible during training. Additionally, the global feature propagation in graphs heavily depends on local connectivity patterns and overall topological structure. The sensitivity of low-degree nodes to adversarial attacks further amplifies this dependency. When low-degree nodes are attacked or subjected to noise [23], [33], [34], [35], [36], their limited neighbors can amplify erroneous information, disrupting the global consistency of feature distribution and degrading the model's performance on node classification tasks.

Various strategies have been proposed to mitigate these issues. One common approach is to increase the depth of GCNs to extend the range of message propagation. Although the strategy can alleviate some misalignment issues, it also introduces the problem of over-smoothing, where node representations become indistinguishable, reducing classification accuracy. Furthermore, long-range information often becomes overly compressed during propagation, further impacting performance [22], [32]. Another approach involves designing auxiliary node generators to create bridging nodes that connect inaccessible labeled and unlabeled nodes, improving distribution alignment [37], [38]. However, this method has notable limitations. The generated nodes may alter the original spatial layout of the graph, and their features may lack semantic relevance, making it harder for the model to interpret their contributions to classification. Moreover, the performance of this approach heavily depends on the quality of the node generator, limiting its generalization ability.

To address these challenges, this paper proposes an augmented GCN framework, AGCN-LR, to enhance the reachability of labeled nodes while mitigating the over-smoothing and feature compression issues caused by deeper GCN architectures. Without altering the original spatial layout of nodes, AGCN-LR introduces multiple augmented graphs to establish direct connections between labeled nodes and low-degree unlabeled nodes, significantly improving labeled node reachability and aligning the distributions of labeled and unlabeled nodes. Additionally, AGCN-LR incorporates a contrastive learning loss and an attention-based fusion mechanism to maintain embedding consistency across different augmented graphs and perform weighted fusion of these embeddings, further enhancing robustness. By sequentially integrating these components, AGCN-LR not only improves labeled node reachability, but also enhances the model's robustness when dealing with sparse label information and graph noise.

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

(1) We propose an augmented GCN framework (AGCN-LR), which is a plug-and-play framework that can be easily integrated into most GCNs to improve their performance and robustness.
(2) AGCN-LR constructs multiple augmented graphs and performs information fusion under consistency constraints, significantly improving the reachability of labeled nodes while enhancing its adaptability in sparse label and graph noise scenarios.
(3) Conduct extensive experiments on four benchmark datasets (Cora, Citeseer, Pubmed, ACM), demonstrating that AGCN-LR outperforms state-of-the-art GCN models in semi-supervised node classification, and exhibits stronger advantages in sparse label and noisy graph scenarios.

The remainder of this paper is organized as follows: Section II reviews related work. The proposed AGCN-LR framework is described in Section III. Experimental results and analysis are presented in Section IV. Finally, Section VI concludes the study.

II. RELATE WORK

A. GRAPH NEURAL NETWORKS FOR SEMI-SUPERVISED NODE CLASSIFICATION

Graph Neural Networks (GNNs), as a class of deep learning models specifically designed for graph-structured data, have demonstrated significant potential across various domains in recent years [1], [2], [3], [6], [19], [20], [21], [30], [33], [39]. By capturing complex relationships and dependencies among nodes, GNNs effectively encode and utilize graph structural information, achieving widespread applications in tasks such as social network analysis, bioinformatics, and recommendation systems [40], [41]. Recent advancements in graph augmentation techniques and contrastive learning methods have shown promise in mitigating these challenges by improving the representation learning process, enabling better alignment of node distributions [42], [43], [44]. Among these applications, semi-supervised node classification is particularly critical, aiming to infer the labels of a large number of unlabeled nodes using limited labeled node information [1], [24], [27], [30].

Among the various GNN models, Graph Convolutional Networks (GCNs) are one of the most widely adopted representatives, excelling in semi-supervised node classification due to their efficient message-passing mechanisms and feature extraction capabilities. By aggregating feature information from a node's neighbors and combining it with the node's own features, GCNs effectively encode both local and global structural information of graphs. Stacking multiple GCN layers enables labeled nodes to gather information from more distant neighbors, capturing the distributional properties of unlabeled nodes more effectively. However, GCNs also face significant challenges in practical applications [32], [34], [45], [46], [47].

A core issue is the uneven distribution of labeled nodes in the graph, which limits the accessibility of labeled nodes for many unlabeled nodes and restricts the range of information propagation. This imbalance not only hinders the alignment

of labeled and unlabeled nodes but also significantly reduces the classification performance of the model. To address this issue, researchers have proposed various improvements. For instance, increasing the depth of GCN layers extends the range of message propagation, enhancing the model's ability to capture long-range information [31], [32], [48], [49], [50]. However, this strategy introduces over-smoothing issues, where node representations become overly similar, leading to degraded classification performance. Moreover, long-range information is often excessively compressed during propagation, further impacting model performance [51].

Another approach involves designing auxiliary node generators to create bridging nodes that connect inaccessible labeled nodes with unlabeled nodes, thereby improving distribution alignment [37], [52], [53]. While this strategy enhances the reachability of labeled nodes to some extent, it often alters the spatial layout of the graph. Additionally, generated nodes may lack meaningful semantic information, making it harder for the model to interpret their contributions to classification. The effectiveness of this method heavily depends on the quality of the node generator, limiting its generalizability.

To address these issues, this paper proposes an augmented GCN-based solution, AGCN-LR. Without altering the original spatial layout of nodes, AGCN-LR constructs multiple augmented graph structures to directly connect labeled nodes with low-degree unlabeled nodes, significantly enhancing labeled node reachability. At the same time, it effectively mitigates over-smoothing and information compression issues caused by increased GCN depth.

B. AUGMENTATION GRAPHS

In recent years, the design of augmented graph structures has gained widespread attention in the field of graph data processing, particularly for improving the performance of node classification tasks [10], [23], [38], [54], [55]. By diversifying the structural properties of the original graph, augmented graphs can enrich the connectivity between nodes and enhance the model's ability to learn both global and local features of the graph. Unlike traditional graph structures, which rely solely on neighborhood relationships, augmented graphs introduce new edges or design alternative views that explicitly connect distant nodes or supplement latent semantic associations. This improves the model's generalization capability and adaptability to complex structures.

It is worth noting that most existing studies focus primarily on using augmented graph structures to improve global feature capture and enhance model robustness. However, there has been relatively little direct exploration of enhancing label reachability to address distribution misalignment. Augmented graphs, by introducing supplementary edges or reconstructing local structures, can explicitly connect distant or low-degree nodes without altering the original spatial layout of nodes [10], [11], [38], [54]. This capability provides a novel approach to address the distribution alignment challenges caused by the uneven distribution of labeled nodes in GCNs.

In addition, augmented graphs can be used to capture self-supervised information [55], [56]. By designing a suitable contrastive learning paradigm, consistency constraints are applied to capture the latent consistency information between the original and augmented graphs. This approach can be used not only in self-supervised learning but also in semi-supervised learning. Inspired by this, we introduce the concept of contrastive learning into the proposed framework, further enhancing the capture of relational information between augmented graphs, thereby improving the robustness of the learning process. This addition helps the model better generalize to diverse learning scenarios, ensuring stronger performance in both sparse labeling and noisy graph scenarios.

III. THE PROPOSED FRAMEWORK: AGCN-LR

A. GCN ENCODER

Consider an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ represents a set of n nodes [57], [58]. The node features are captured by a matrix $\boldsymbol{X} \in \mathbb{R}^{N \times F}$, with N indicating the total number of nodes and f representing the feature dimensionality. The adjacency matrix $\boldsymbol{A} \in \{0, 1\}^{N \times N}$ encodes the connectivity between nodes, where $a_{ij} = 1$ indicates a connection between nodes v_i and v_j. The degree matrix associated with $\boldsymbol{A}$ is denoted as $\boldsymbol{D}$. To incorporate self-loops, we define the adjusted adjacency matrix $\tilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}_N$ and its corresponding degree matrix $\tilde{\boldsymbol{D}} = \boldsymbol{D} + \boldsymbol{I}_N$, where $\boldsymbol{I}_N$ is the identity matrix [59]. The symmetrically normalized adjacency matrix with self-loops is represented as $\hat{\boldsymbol{A}} = \tilde{\boldsymbol{D}}^{-\frac{1}{2}} \tilde{\boldsymbol{A}} \tilde{\boldsymbol{D}}^{-\frac{1}{2}}$. Let C denote the number of node classes.

The vanilla GCN, introduced by Kipf and welling [1], computes node representations by aggregating neighborhood information using a symmetrically normalized adjacency matrix $\hat{\boldsymbol{A}}$. The basic formulation of the GCN model is expressed as:

$$\boldsymbol{\mathcal{Z}}_{GCN} = \sigma(\hat{\boldsymbol{A}} ReLU(\hat{\boldsymbol{A}} \boldsymbol{X} \boldsymbol{W}^{(0)}) \boldsymbol{W}^{(1)}) \tag{1}$$

Here, $\boldsymbol{W}^{(0)}$ and $\boldsymbol{W}^{(1)}$ are the learnable weight matrices for the first and second layers of the GCN, respectively [60], [61]. The non-linear activation functions σ and $ReLU$ introduce non-linearity [62], [63]. $\boldsymbol{\mathcal{Z}}_{GCN}$ represents the final output of a two-layer GCN.

B. FRAMEWORK OVERVIEW

This section introduces the Augmented Graph Convolutional Network for enhancing Labeled Reachability (AGCN-LR). The architecture of AGCN-LR is shown in Figure 1. Our framework encompasses the following key parts.

(1) **Augmentation Graph Part for Enhancing Label Node Reachability**: This part aims to enhance the reachability of labeled nodes for low-degree nodes. This strengthens the reachability of labeled nodes and

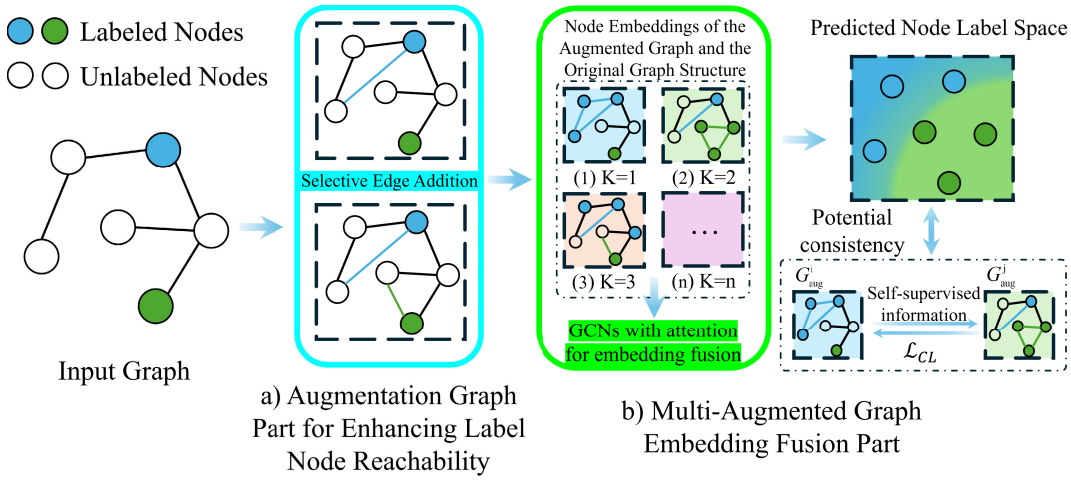


FIGURE 1. The framework of AGCN-LR.

improves the propagation of labeled node information to unlabeled nodes.

(2) **Multi-Augmented Graph Embedding Fusion Part**: This part leverages Graph Convolutional Networks (GCNs) to process the augmented graphs. Multiple embeddings from augmented graphs are fused through an attention mechanism to create consistent low-dimensional representations, ensuring robust node classification despite sparse labeled data.

C. AUGMENTATION GRAPH PART FOR ENHANCING LABEL NODE REACHABILITY

To effectively explore the potential connections between low-degree nodes and labeled nodes, we generate a similarity distribution between them based on the feature similarity of nodes and the shortest path length to the labeled nodes.

$$S(v,u) = \alpha \cdot \mathrm{Sim}(\mathbf{x}_v, \mathbf{x}_u) + \beta \cdot \frac{1}{\mathrm{SP}(v,u)} \tag{2}$$

where $S(v,u)$ represents the similarity between the low-degree node v and the labeled node u. $\mathbf{x}_v$ and $\mathbf{x}_u$ are the feature vectors of node v and node u, respectively. $\mathrm{Sim}(\mathbf{x}_v, \mathbf{x}_u)$ denotes the similarity between the feature vectors (e.g., cosine similarity). $\mathrm{SP}(v,u)$ denotes the shortest path length between node v and node u. α and β are hyperparameters that control the contribution of these two components, where $\alpha + \beta = 1$.

Immediately, we randomly sample edges between low-degree nodes and labeled nodes based on the similarity distribution S, thereby generating multiple augmentation graphs.

$$\mathcal{E}^k_{\mathrm{aug}} = \{(v,u) \mid v \in \mathcal{V}_L, u \in \mathcal{V}_H, p(v,u)^k \sim S(v,u)\} \tag{3}$$

$$\mathcal{G}^k_{\mathrm{aug}} = (\mathcal{V}, \mathcal{E} \cup \mathcal{E}^k_{\mathrm{aug}}) \tag{4}$$

where $\mathcal{E}^k_{\mathrm{aug}}$ represents the k-th set of edges randomly sampled between the set of low-degree nodes $\mathcal{V}_L$ and the set of labeled nodes $\mathcal{V}_H$, with the sampling probability $p(v,u)^k$ following the similarity distribution $S(v,u)$. $\mathcal{G}^k_{\mathrm{aug}}$ denotes the k-th augmentation graph, where $\mathcal{V}$ is the set of nodes, $\mathcal{E}$ is the original edge set, and $\mathcal{E}_{\mathrm{aug}}$ is the newly added edge set.

D. MULTI-AUGMENTED GRAPH EMBEDDING FUSION PART

To learn the generated augmented graphs, we introduce a graph embedding fusion framework, which consists of two parts: 1) graph embedding fusion; 2) graph semantic consistency learning.

1) GRAPH EMBEDDING FUSION

First, learning the generated augmented graphs and the original graph embeddings by using GCNs. Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we take vanilla GCN as an encoder to learn graph embedding. We input $\mathcal{G}$ into the two-layer GCN as follows:

$$\boldsymbol{\mathcal{Z}}^k = (\hat{\boldsymbol{A}}^k ReLU(\hat{\boldsymbol{A}}^k \boldsymbol{X} \boldsymbol{W}^{(0)}) \boldsymbol{W}^{(1)}) \tag{5}$$

where $\boldsymbol{\mathcal{Z}}^k$ denotes the embedding result learned from the k-th input graph.

To efficiently fuse the embeddings generated by the augmented graphs and the original graph, we incorporate an attention mechanism [64] to obtain a more comprehensive semantic representation.

With K embeddings, namely $\boldsymbol{\mathcal{Z}}^1$, $\boldsymbol{\mathcal{Z}}^2$, ..., and $\boldsymbol{\mathcal{Z}}^K$, we use attention mechanisms to determine their respective importance.

$$(\alpha_1, \alpha_2, \ldots, \alpha_m) = att(\boldsymbol{\mathcal{Z}}^1, \boldsymbol{\mathcal{Z}}^2, \ldots, \boldsymbol{\mathcal{Z}}^K) \tag{6}$$

where $\alpha_1, \alpha_2, \ldots, \alpha_m \in \mathbb{R}^{N\times 1}$ indicate the attention values of N nodes with embeddings $\boldsymbol{\mathcal{Z}}^1, \boldsymbol{\mathcal{Z}}^2, \ldots, \boldsymbol{\mathcal{Z}}^K$, respectively. $\alpha_{ij} = softmax(w_{ij})$, $w_{ij} = \mathbf{q}_i^T \cdot tanh(\boldsymbol{W}_{att} \cdot (\boldsymbol{z}_j)^T + \mathbf{b}_i)$, the unnormalized attention value w_{ij} for j-th node in embedding matrice $\boldsymbol{\mathcal{Z}}^i$ through a shared attention vector $\mathbf{q}_i \in \mathbb{R}^{F'\times 1}$,

$\boldsymbol{W}_{att} \in \mathbb{R}^{F^h \times F'}$ is the weight matrix, and $\mathbf{b}_i \in \mathbb{R}^{F' \times 1}$ is the bias vector.

Finally, the new embedding ($\boldsymbol{\mathcal{Z}}^{fusion}$ is aggregated by the attention mechanism with the attention values α.

$$\boldsymbol{\mathcal{Z}}^{\text{fusion}} = \sum_{k=1}^{K} \alpha_i \cdot \boldsymbol{\mathcal{Z}}^k \tag{7}$$

where $\boldsymbol{\mathcal{Z}}^k$ represents the embedding of the k-th graph, α_k is the weight for the k-th embedding computed by the attention mechanism, and $\boldsymbol{\mathcal{Z}}^{\text{fusion}}$ is the fused graph embedding.

2) GRAPH SEMANTIC CONSISTENCY LEARNING

Considering the potential consistency among multiple augmented graphs, we introduce a graph contrastive loss [23], [65] to capture the self-supervised information hidden between the augmented graphs. Following the InfoNCE objective in contrastive learning [66], for each node, we regard $\left(z_n^k, z_n^{k'}\right)$ as the positive pair, and $\left\{\left(z_n^k, z_{n'}^{k'}\right)\right\}_{[n' \neq n]}$ as the negative pairs. Then the contrastive loss $\mathcal{L}_{CL}$ for K augmentations is given by:

$$\mathcal{L}_{CL} = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{K^2}\sum_{k=1}^{K}\sum_{k'=1}^{K} -\log\frac{\exp\left(z_n^{k'\top} z_n^k/\tau\right)}{\sum_{n'=1}^{N}\exp\left(z_{n'}^{k'\top} z_n^k/\tau\right)} \tag{8}$$

where we use inner-product scaled by a temperature hyper-parameter τ as the similarity estimator.

In semi-supervised node classification tasks. The optimization objective is commonly formulated as:

$$\mathcal{L}_O = -\sum_{v_k \in \mathcal{V}_L}\sum_{j}^{C} \boldsymbol{Y}_{kj}^L \log(\boldsymbol{\mathcal{Z}}_{kj}^{fusion}) \tag{9}$$

where $\mathcal{L}_O$ denotes the cross-entropy loss. $\boldsymbol{Y}^L$ denotes the real label set (one-hot encoder).

Therefore, the final loss function of the model is expressed as follows:

$$\mathcal{L} = (1-\lambda)\mathcal{L}_O + \lambda\mathcal{L}_{CL} \tag{10}$$

where λ is a balance hyper-parameter.

Algorithm 1 shows the entire training process of the model.

Algorithm 1 The Training Framework of AGCN-LR

Input: input graph $\mathcal{G}$; epochs T; hyper-parameters K, τ, λ;
Output: The sample class prediction matrix $\boldsymbol{\mathcal{Z}}^{\text{fusion}}$;

```
Initialize hyper-parameters: K, τ, λ;
Generating K augmented graphs {G^1, G^2, ..., G^K} via Eqs.(2)-(4)
for int epochs = 0 to T − 1 do
    Calculate embedding Z^k by Eq.(1)
    Calculate Z^fusion by Eqs.(5)-(7);
    Calculate L_CL, L_O respectively by Eq.(8), and Eq.(9);
    Calculate total loss: L = (1 − λ)L_O + λL_CL;
end for
return Z^fusion
```

IV. EXPERIMENTS

In this section, we evaluate our proposed model by comparing it with several well-known graph-based semi-supervised node classification models. We also conduct additional experiments to assess the impact of hyperparameters and the performance of the components of AGCN-LR. The visualization of model accuracy further highlights the effectiveness of AGCN-LR.

A. DATASETS

In this study, we utilize four real benchmark datasets for semi-supervised classification tasks. Table 1 summarizes the characteristics of the four datasets used in the experiments. In these datasets, nodes represent publications and edges represent citation links. Cora, Citeseer, and Pubmed [67], [68], [69] are widely used citation network benchmark datasets, while ACM [19] is collected from the ACM digital library. In these networks, each node represents a paper, and an edge indicates a citation from one paper to another. The node features contain relevant information about the papers, and the node labels correspond to the academic topics of the papers.

TABLE 1. Description of the dataset for the semi-supervised classification tasks.

No.	Dataset	Node	Edges	Features	Classes
1	Cora	2708	5429	1433	7
2	Citeseer	3327	4732	3703	6
3	Pubmed	19717	443328	500	3
4	ACM	3025	13128	1870	3

B. BASELINES

To validate the effectiveness of our approach, we select 8 state-of-the-art GNN methods as baselines for comparison:

- **GCN** [1]: A semi-supervised graph convolutional network model that learns node representations by aggregating information from neighboring nodes.
- **GAT** [2]: A graph convolutional network model based on attention mechanisms used to aggregate node features.
- **SGC** [3]: A simplified graph convolutional network model that directly propagates through the adjacency matrix multiple times and removes nonlinear activation functions for improved efficiency.
- **MOGCN** [29]: A graph convolutional network model that pre-constructs multi-order adjacency matrices, builds individual simple GCNs on each specific order adjacency matrix, and merges the results of each GCN using ensemble learning.

•**LaenNet** [18]: A graph neural network that utilizes label-enhancement and neighborhood attention mechanisms to improve node classification performance.

•**PA-GCN** [23]: A perturbation-augmented graph convolutional network method that integrates graph self-supervised learning and semi-supervised learning.

•**N-GCN** [27]: A graph convolutional network method that captures multi-order neighborhood features of nodes by constructing multiple channels.

•**IGCL** [11]: A contrastive learning-based graph neural network method that enhances the consistency and robustness of node embeddings through contrastive learning on different views.

•**MAGCN** [28]: A multi-view graph convolutional network method that leverages an attention mechanism to capture multi-scale information and enhance node classification performance across different views.

•**GraphSAGE-GCN** [61]: A graph convolutional network (GCN) that incorporates the GraphSAGE framework to sample and aggregate neighbor information dynamically. By leveraging different aggregation functions (mean, LSTM, pooling), it efficiently scales to large graphs while improving node classification performance across diverse graph structures.

C. EXPERIMENT SETTINGS

In our study, the model is optimized using the Adam algorithm, and the experimental results are evaluated using classification accuracy (ACC) as the performance metric. For consistency, all models are configured with the following parameters: the number of iterations is set to 500, the learning rate is set to 0.003, and the dropout rate is set to 0.5. Hyperparameter λ is selected from the range $\{0.00, 0.05, 0.10, \ldots, 0.95\}$, while τ is chosen from $\{0.1, 0.2, \ldots, 1.0\}$. The number of augmented graphs K is configured to one of $\{1, 2, 3, 4, 5\}$. These hyperparameters are optimized to achieve the best performance, and the model's performance is reported based on the optimal configuration. For other baseline methods, hyperparameter values are adopted from their respective papers or the default settings provided in the corresponding code implementations. To ensure stability and reliability, this study performs 50 repeated experiments and reports the mean performance, as the mean across multiple runs better reflects the robustness of the models.

D. ARCHITECTURE STUDY AND LABEL RATE IMPACT ANALYSIS

This study combines AGCN-LR with the GCN, GAT, and SGC models, resulting in three new models: GCN+AGCN-LR, GAT+AGCN-LR, and SGC+AGCN-LR. These models are then compared with the baseline models at various label rates. The label rates for the Cora and Citeseer datasets range from 0.05% to 5.20% and 0.05% to 3.60%, respectively, while the label rates for the Pubmed and ACM datasets range from 0.08% to 0.30% and 0.50% to 2.00%, respectively. The average classification accuracy (ACC) for 1000 test nodes is reported, with the experimental results summarized in Table 2. The results demonstrate that the average classification accuracy of AGCN-LR significantly outperforms the baseline models (GCN, GAT, and SGC) across different datasets and label rates. Notably, AGCN-LR performs exceptionally well at low label rates, showcasing its strong adaptability and robustness in sparse label scenarios.

In addition, we conducted a time complexity analysis experiment. Table 3 compares the total training time of AGCN-LR with GCN, GAT, SGC, MAGCN and GraphSAGE-GCN on the three most commonly used benchmark datasets (Cora, Citeseer, Pubmed), where MAGCN and GraphSAGE-GCN are two comparison baselines similar to AGCN-LR. The table shows that although the training time of AGCN-LR is slightly higher than that of basic models like GCN, GAT, and SGC, it still demonstrates a significant time-efficiency advantage compared to the latest benchmark models, such as MAGCN and GraphSAGE-GCN.

E. COMPARATIVE ANALYSIS OF CLASSIFICATION RESULTS

In the experiments described below, we use the standard label rates of 5.20% for Cora, 3.60% for Citeseer and 2.00% for Pubmed and ACM.

Under the label rates, the semi-supervised classification results of the baseline models and optimal hyperparameters are shown in Table 4.

Overall, AGCN-LR outperforms other baseline models in terms of classification accuracy across all datasets. AGCN-LR demonstrates significant classification performance advantages on different datasets. Furthermore, compared to models like LaenNet, PA-GCN, MAGCN, and GraphSAGE, which also address label reachability, AGCN-LR outperforms these models on most datasets, with a particularly notable advantage on the Citeseer dataset. These results validate the effectiveness of the AGCN-LR approach.

1) CLASSIFICATION RESULT VISUALIZATION

Figure 2 show the t-SNE [70] visualization results of 1000 test samples on the Cora and Citeseer datasets. By mapping high-dimensional data to a 2D space, we can visually observe the classification differences between the two models.

From Figure 2, it is evident that the GCN+AGCN-LR model outperforms GCN on both the Cora and Citeseer datasets. RLGCN exhibits stronger node clustering within the same class, and clearer class boundaries. This demonstrates that AGCN-LR better captures the direct relationships between labeled and unlabeled nodes, improving classification performance on complex graph data.

2) PARAMETER SENSITIVITY STUDY

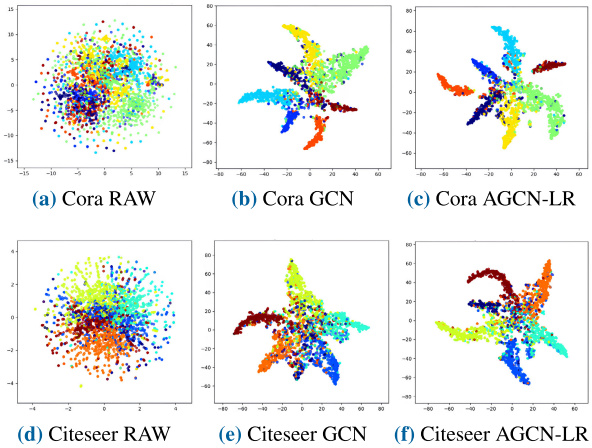
First, we perform a sensitivity analysis of λ across four datasets. We train independent AGCN-LR instances with different λ values (ranging from 0.1 to 0.9 with a step size of 0.1) and evaluate the average classification accuracy (ACC), with the results shown in Figure 3. It is observed that ACC peaks

TABLE 2. Accuracy(%) of node classification tasks. The best results are highlighted.

	Label rate	GCN	GAT	SGC	MOGCN	LaenNet	PA-GCN	N-GCN	IGCL	MAGCN	GraphSAGE-GCN	AGCN-LR
Cora	0.05%	44.71	44.35	45.61	46.10	47.56	46.22	47.03	48.55	47.48	47.31	**49.91**
	0.10%	63.56	66.55	67.30	68.30	69.82	68.52	70.16	71.03	71.32	71.28	**77.19**
	5.20%	81.27	83.44	81.58	83.12	83.53	82.20	83.37	80.18	83.46	81.36	**84.27**
Citeseer	0.05%	43.56	58.78	59.92	61.17	63.24	61.57	62.12	63.54	63.48	63.57	**67.58**
	0.10%	55.09	61.35	62.84	64.12	66.52	64.33	66.11	65.45	66.31	65.62	**70.25**
	3.60%	70.29	72.12	70.83	72.64	73.23	72.95	72.36	70.52	71.67	72.47	**74.05**
Pubmed	0.08%	56.98	58.31	59.20	60.84	62.17	61.45	61.72	62.50	62.49	62.87	**65.94**
	0.15%	64.02	70.45	71.13	72.50	73.35	72.73	73.02	74.13	74.28	73.95	**76.68**
	0.30%	79.56	79.38	80.24	80.07	80.26	80.34	80.14	78.25	79.54	80.32	**81.25**
ACM	0.50%	65.12	69.62	70.14	71.55	72.62	71.30	72.48	73.36	72.81	73.56	**75.31**
	1.00%	79.45	81.68	80.32	83.10	84.10	83.24	83.37	82.52	82.68	83.73	**85.08**
	2.00%	87.34	87.67	86.45	88.74	**90.23**	87.38	89.46	86.64	87.38	89.53	90.12

TABLE 3. Summary of total training time (in seconds) across the three most common benchmark datasets.

Method	Cora	Citeseer	Pubmed
Advanced GCNs			
MAGCN [28]	4.35	6.17	74.46
GraphSAGE-GCN [61]	4.86	8.97	112.03
GCNs and AGCN-LR enhanced GCNs			
GCN [1]	0.72	0.87	11.34
GCN+AGCN-LR	1.46	2.15	23.38
GAT [2]	1.45	3.78	18.21
GAT+AGCN-LR	3.16	7.24	37.13
SGC [3]	0.11	0.18	0.25
SGC+AGCN-LR	0.22	0.39	0.62



(a) Cora RAW (b) Cora GCN (c) Cora AGCN-LR

(d) Citeseer RAW (e) Citeseer GCN (f) Citeseer AGCN-LR

FIGURE 2. The T-SNE visualization results of the Cora and Citeseer datasets.

within the range of [0.4, 0.6], suggesting that higher λ values lead the model to overly rely on structural information while

TABLE 4. Node classification accuracy (%). The best results are highlighted.

Model	Cora	Citeseer	Pubmed	ACM
GCN	81.27	70.29	79.56	87.34
GAT	83.44	72.12	79.38	87.67
SGC	81.58	70.83	80.24	86.45
MOGCN	83.12	72.64	80.57	88.74
LaenNet	83.53	73.23	80.86	**90.23**
PA-GCN	82.20	72.95	80.34	87.38
N-GCN	83.37	72.36	80.14	89.46
IGCL	80.18	70.52	78.25	86.64
MAGCN	83.46	71.67	79.54	87.38
GraphSAGE-GCN	81.36	72.47	80.32	89.53
GCN+AGCN-LR	84.27	74.05	81.25	90.12
GAT+AGCN-LR	**84.43**	**75.32**	**81.74**	89.68
SGC+AGCN-LR	83.86	74.05	80.68	89.24

neglecting label information, which can cause overfitting or underfitting during training and negatively impact the final classification accuracy. On the other hand, lower λ values fail to fully utilize the graph structure, preventing the model from capturing the complex relationships within the data. Therefore, selecting an appropriate λ is crucial for balancing supervision and structural information, which improves the model's generalization ability.

Next, we investigate the effect of the augmentation graph number K on model performance using the Cora and Citeseer datasets. As shown in Table 5, ACC is maximized at $K = 2$, and increasing K beyond this threshold leads to a decline in performance, while $K = 1$ fails to enhance connectivity effectively, limiting label propagation. As K increases, the richness of the graph structure improves,

but excessive augmentation introduces noise, disrupting information transfer and reducing accuracy. In contrast, $K = 1$ results in a simpler graph structure, unable to connect low-degree and labeled nodes effectively, thus limiting label propagation. Therefore, $K = 2$ achieves the best balance between structure enrichment and noise control.

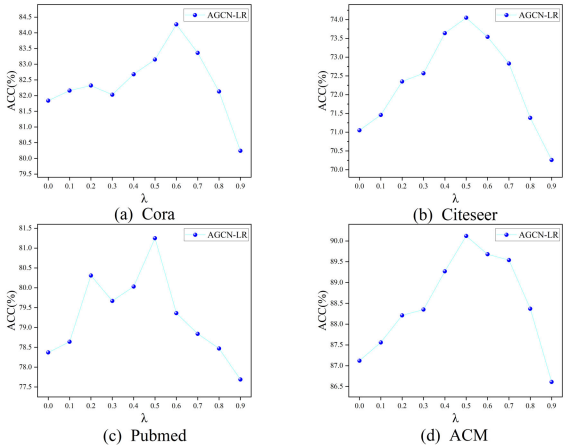


FIGURE 3. Classification results of AGCN-LR under different hyperparameter values.

TABLE 5. Impact of augmentation graph number K on classification accuracy.

Dataset	K = 1	K = 2	K = 3	K = 4
Cora	82.1	**84.3**	83.5	82.7
Citeseer	72.3	**73.8**	73.1	72.4

3) GRAPH NOISE SENSITIVITY ANALYSIS

In this subsection, we generated multiple synthetic datasets based on three commonly used benchmark datasets (Cora, Citeseer, Pubmed). These synthetic datasets were designed with varying levels of noise to simulate common real-world scenarios. The noise levels were set using random edge reverse attacks with noise ratios of {0.2, 0.4, 0.6, 0.8}. On these synthetic datasets, we evaluated the performance of AGCN-LR and graph convolutional networks (GCN, GAT, SGC) on the semi-supervised node classification task, aiming to compare and analyze the sensitivity of baseline methods and AGCN-LR to graph noise. The results are summarized in Figures 4-5.

From the experimental results, we can draw the following conclusions:

(1) AGCN-LR enhances the robustness of base models: For example, when $\alpha = 0.8$, GAT+AGCN-LR improves performance on Pubmed by 144.06% compared to GAT. This improvement can be attributed to the augmented graph's ability to identify potential noisy edges. By adjusting the graph structure, AGCN-LR ensures the reachability of labeled nodes while simultaneously removing potential noisy edges.

(2) AGCN-LR enables smoother performance degradation as noise increases for GCNs: This can be attributed to AGCN-LR's graph mutual learning mechanism, which compensates for the loss of effective information in common noisy environments.

4) ABLATION ANALYSIS

This section analyzes the performance of each component in AGCN-LR. To do so, we design two model variants: one based on the GCN learning part and one without the graph contrastive loss in AGCN-LR.

- **GCN**: This model removes the augmentation graph part and focuses only on learning original graph structural information using the GCN model.
- **w/o**: This model removes consistent contextual learning in the augmentation graph learning, meaning it does not include the graph contrastive loss in AGCN-LR.

The classification accuracy of these variants is shown in Table 6. It can be observed that the augmentation graph part can improve the label reachability of low-degree nodes, thereby significantly enhancing the performance of RLGNNs. Notably, when the augmentation graph part does not incorporate consistent contextual learning, the classification accuracy on some datasets actually performs worse than the baseline GCN model. This is becauseph contrastive loss helps guide the model to focus more on similar contextual relations, preventing attention from becoming too dispersed, and ultimately improving the label accessibility of similarly labeled nodes.

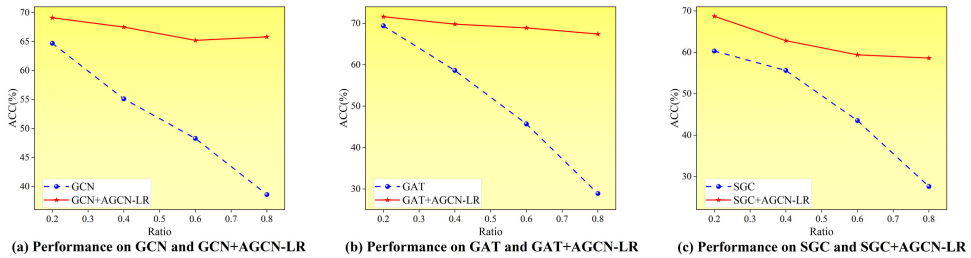
TABLE 6. Classification accuracy of AGCN-LR variants.

Model	Cora	Citeseer	Pubmed	ACM
GCN	81.27	70.29	79.56	87.34
w/o	83.84	72.05	79.37	89.12
GCN+AGCN-LR	**84.27**	**74.05**	**81.25**	**90.12**

V. DISCUSSION

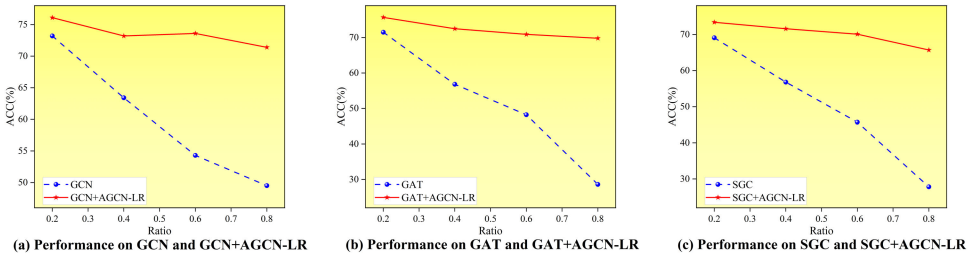
AGCN-LR improves the reachability of labeled nodes by introducing multiple augmented graphs, particularly enhancing the connections between low-degree unlabeled nodes and labeled nodes. The method combines contrastive learning loss and attention mechanisms to ensure consistency of embeddings generated from different augmented graphs. By performing weighted fusion of these embeddings, AGCN-LR effectively avoids the over-smoothing and feature compression issues commonly found in deep GCN models, without increasing the number of layers or introducing auxiliary nodes.

Although AGCN-LR demonstrates strong robustness and generalization ability in semi-supervised learning tasks, it still has some limitations.



(a) Performance on GCN and GCN+AGCN-LR (b) Performance on GAT and GAT+AGCN-LR (c) Performance on SGC and SGC+AGCN-LR

FIGURE 4. Test performance of models on Citeseer under random edge removal attack.



(a) Performance on GCN and GCN+AGCN-LR (b) Performance on GAT and GAT+AGCN-LR (c) Performance on SGC and SGC+AGCN-LR

FIGURE 5. Test performance of models on Pubmed under random edge removal attack.

1) AGCN-LR heavily depends on the quality of the input data, and incorrect handling of noise, missing values, or incomplete features can result in suboptimal performance.
2) Both the original and augmented graphs have static structures, which limits their adaptability and flexibility in certain dynamic learning scenarios. Therefore, further research on dynamic graph structure learning strategies to enhance the adaptability of AGCN-LR is meaningful.
3) The learning paradigm of the mutual convolution layer is still based on the homogeneity assumption. If there are significant semantic differences between the original and augmented graphs, the model's performance may not be ideal. It is necessary to explore the learning of heterogeneous interaction information.

VI. CONCLUSION

In this paper, we propose a novel augmented GCN scheme, AGCN-LR, for semi-supervised node classification tasks. This scheme introduces multiple augmented graphs to enhance the direct connections between low-degree unlabeled nodes and labeled nodes, significantly improving the reachability of labeled nodes. Furthermore, AGCN-LR incorporates a contrastive learning loss function and an attention mechanism, ensuring consistency across embeddings generated from different augmented graphs. It also performs weighted fusion of these embeddings, effectively mitigating the over-smoothing and feature compression issues caused by stacking GCN layers, without increasing the number of layers or introducing auxiliary nodes.

Experimental results demonstrate that AGCN-LR achieves superior classification performance compared to other state-of-the-art GCN models on four benchmark datasets: Cora, Citeseer, Pubmed, and ACM. These results highlight the strong generalizability of AGCN-LR. By optimizing the reachability of labeled nodes, AGCN-LR effectively addresses the challenges faced by traditional GCN models, such as oversmoothing and feature compression caused by deeper architectures. Additionally, it enhances the performance of existing GCN models while preserving the original graph layout, delivering significant improvements in semi-supervised node classification tasks.

For future work, our goal is to optimize the augmented graph construction strategy in AGCN-LR by addressing three key challenges: improving data robustness, improving adaptability in dynamic scenarios, and refining heterogeneous interaction learning. First, we plan to explore more optimized graph generation techniques, such as sparse matrix representations and hardware acceleration, to ensure that AGCN-LR remains scalable in large datasets. Second, we will investigate dynamic graph learning methods, including incremental learning and reinforcement learning-based structure adaptation, to enable AGCN-LR to adjust to evolving data distributions [71]. Third, we intend to improve heterogeneous interaction learning using multi-view and meta-learning techniques to better capture complex relationships.Furthermore, to enhance interpretability, we plan to incorporate explainability techniques such as SHAP [72] and LIME [73] to analyze layer-wise contributions in AGCN-LR.

REFERENCES

[1] T. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. 5th Int. Conf. Learn. Represent.*, Jan. 2016, pp. 1–14.
[2] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lió, and Y. Bengio, "Graph attention networks," in *Proc. 6th Int. Conf. Learn. Represent.*, Jan. 2017, pp. 1–12.

[3] F. Wu, T. Zhang, A. H. Souza, C. Fifty, T. Yu, and K. Q. Weinberger, "Simplifying graph convolutional networks," in *Proc. 36th Int. Conf. Mach. Learn.*, Jan. 2019, pp. 6861–6871.

[4] B. Tang, X. Chen, S. Wang, Y. Xuan, and Z. Zhao, "Generalized heterophily graph data augmentation for node classification," *Neural Netw.*, vol. 168, pp. 339–349, Nov. 2023.

[5] J. Zhang, J. Huang, J. Gao, R. Han, and C. Zhou, "Knowledge graph embedding by logical-default attention graph convolution neural network for link prediction," *Inf. Sci.*, vol. 593, pp. 201–215, May 2022.

[6] P. Zhang, J. Chen, C. Che, L. Zhang, B. Jin, and Y. Zhu, "IEA-GNN: Anchor-aware graph neural network fused with information entropy for node classification and link prediction," *Inf. Sci.*, vol. 634, pp. 665–676, Jul. 2023.

[7] N. Ziraki, F. Dornaika, and A. Bosaghzadeh, "Multiple-view flexible semi-supervised classification through consistent graph construction and label propagation," *Neural Netw.*, vol. 146, pp. 174–180, Feb. 2022.

[8] T. Ma, H. Wang, L. Zhang, Y. Tian, and N. Al-Nabhan, "Graph classification based on structural features of significant nodes and spatial convolutional neural networks," *Neurocomputing*, vol. 423, pp. 639–650, Jan. 2021.

[9] M. Khademi and O. Schulte, "Deep generative probabilistic graph neural networks for scene graph generation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, Apr. 2020, pp. 11237–11245.

[10] J. Duan, S. Wang, P. Zhang, E. Zhu, J. Hu, H. Jin, Y. Liu, and Z. Dong, "Graph anomaly detection via multi-scale contrastive learning networks with augmented view," in *Proc. AAAI Conf. Artif. Intell.*, vol. 37, Jun. 2023, pp. 7459–7467.

[11] H. Liang, X. Du, B. Zhu, Z. Ma, K. Chen, and J. Gao, "Graph contrastive learning with implicit augmentations," *Neural Netw.*, vol. 163, pp. 156–164, Jun. 2023.

[12] X. Wang, X. Yang, P. Wang, H. Yu, and T. Xu, "SSGCN: A sampling sequential guided graph convolutional network," *Int. J. Mach. Learn. Cybern.*, vol. 15, no. 5, pp. 2023–2038, May 2024.

[13] K. Sun, Z. Lin, and Z. Zhu, "Multi-stage self-supervised learning for graph convolutional networks on graphs with few labeled nodes," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, Apr. 2020, pp. 5892–5899.

[14] K. Chen and W. Wu, "Class-aware progressive self-training for learning convolutional networks on graphs," *Expert Syst. Appl.*, vol. 238, Mar. 2024, Art. no. 121805.

[15] W. Liu, G. Zhou, X. Mao, S. Bao, H. Li, J. Shi, H. Chen, J. Shen, and Y. Huang, "Global disentangled graph convolutional neural network based on a graph topological metric," *Knowl.-Based Syst.*, vol. 284, Jan. 2024, Art. no. 111283.

[16] N. Jia, X. Tian, T. Yang, S. Li, and L. Jiao, "Self-restrained contrastive enhanced network for graph structure learning," *Expert Syst. Appl.*, vol. 249, Sep. 2024, Art. no. 123520.

[17] J. Bi and F. Dornaika, "Sample-weighted fused graph-based semi-supervised learning on multi-view data," *Inf. Fusion*, vol. 104, Apr. 2024, Art. no. 102175.

[18] C. Zhang, X. Li, H. Pei, Z. Zhang, B. Liu, and B. Yang, "LaenNet: Learning robust GCNs by propagating labels," *Neural Netw.*, vol. 168, pp. 652–664, Nov. 2023.

[19] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu, "Heterogeneous graph attention network," in *Proc. World Wide Web Conf.*, May 2019, pp. 2022–2032.

[20] H. Y. Chang, J. T. Lin, and Y. J. Wu, "Graph neural networks for predicting gene interactions in bioinformatics," *Bioinf. Adv.*, vol. 38, pp. 205–215, Jun. 2020.

[21] Y. Zhong, G. Li, J. Yang, H. Zheng, Y. Yu, J. Zhang, H. Luo, B. Wang, and Z. Weng, "Learning motif-based graphs for drug–drug interaction prediction via local–global self-attention," *Nature Mach. Intell.*, vol. 6, no. 9, pp. 1094–1105, Aug. 2024.

[22] J. Gilmer, S. S. Schoenholz, P. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proc. 34th Int. Conf. Mach. Learn.*, Jan. 2017, pp. 1263–1272.

[23] Q. Guo, X. Yang, F. Zhang, and T. Xu, "Perturbation-augmented graph convolutional networks: A graph contrastive learning architecture for effective node classification tasks," *Eng. Appl. Artif. Intell.*, vol. 129, Mar. 2024, Art. no. 107616.

[24] M. Ghayekhloo and A. Nickabadi, "CLP-GCN: Confidence and label propagation applied to graph convolutional networks," *Appl. Soft Comput.*, vol. 132, Jan. 2023, Art. no. 109850.

[25] M. Peng, X. Juan, and Z. Li, "Label-guided graph contrastive learning for semi-supervised node classification," *Expert Syst. Appl.*, vol. 239, Apr. 2024, Art. no. 122385.

[26] M. Zhang, Z. Cui, M. Neumann, and Y. Chen, "An end-to-end deep learning architecture for graph classification," in *Proc. 32th AAAI Conf. Artif. Intell.*, vol. 32, Apr. 2018, pp. 4438–4445.

[27] S. Abu-El-Haija, A. Kapoor, B. Perozzi, and J. Lee, "N-GCN: Multi-scale graph convolution for semi-supervised node classification," in *Proc. 35th Conf. Uncertainty Artif. Intell.*, Jan. 2019, pp. 841–851.

[28] K. Yao, J. Liang, J. Liang, M. Li, and F. Cao, "Multi-view graph convolutional networks with attention mechanism," *Artif. Intell.*, vol. 307, Jun. 2022, Art. no. 103708.

[29] J. Wang, J. Liang, J. Cui, and J. Liang, "Semi-supervised learning with mixed-order graph convolutional networks," *Inf. Sci.*, vol. 573, pp. 171–181, Sep. 2021.

[30] X. Shen, D. Sun, S. Pan, X. Zhou, and L. Yang, "Neighbor contrastive learning on learnable graph augmentation," in *Proc. 37th AAAI Conf. Artif. Intell.*, vol. 37, Jun. 2023, pp. 9782–9791.

[31] W. Lu, Z. Guan, W. Zhao, Y. Yang, and L. Jin, "NodeMixup: Tackling under-reaching for graph neural networks," in *Proc. AAAI*, Mar. 2024, vol. 38, no. 13, pp. 14175–14183.

[32] F. D. Giovanni, L. Giusti, F. Barbero, G. Luise, P. Lió, and M. M. Bronstein, "On over-squashing in message passing neural networks: The impact of width, depth, and topology," in *Proc. 40th Int. Conf. Learn. Represent.*, Jan. 2023, pp. 7865–7885.

[33] D. Zhu, Z. Zhang, P. Cui, and W. Zhu, "Robust graph convolutional networks against adversarial attacks," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 2, pp. 767–781, Jul. 2022.

[34] Y. Zhu, L. Feng, Z. Deng, Y. Chen, R. Amor, and M. Witbrock, "Robust node classification on graph data with graph and label noise," in *Proc. 38th AAAI Conf. Artif. Intell.*, vol. 38, Mar. 2024, pp. 17220–17227.

[35] S. Xia, S. Zheng, G. Wang, X. Gao, and B. Wang, "Granular ball sampling for noisy label classification or imbalanced classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 4, pp. 2144–2155, Apr. 2023.

[36] J. Tang, X. Wu, and X. Wang, "Adversarial attack and defense strategies for graph neural networks: A survey," *Neurocomputing*, vol. 458, pp. 85–102, Apr. 2021.

[37] J. Liu, Q. Yang, and J. Tang, "Graph generator-enhanced graph neural networks for semi-supervised node classification," *Neural Netw.*, vol. 150, pp. 243–256, Apr. 2022.

[38] J. Lv, K. Song, Q. Ye, and G. Tian, "Semi-supervised node classification via fine-grained graph auxiliary augmentation learning," *Pattern Recognit.*, vol. 137, May 2023, Art. no. 109301.

[39] M. M. Bronstein, J. Bruna, T. Cohen, and P. Velickovic, "Geometric deep learning: Grids, groups, graphs, geodesics, and gauges," *Nature Rev. Phys.*, vol. 3, no. 12, pp. 895–915, Jan. 2021.

[40] J. Jimenez, "Dipoles and streams in two-dimensional turbulence," 2020, *arXiv:2006.10839*.

[41] H. Wu, X. Liang, H. He, X. Wang, and W. Wang, "Graph convolutional recommendation with user-specific hubs," *ACM Trans. Intell. Syst. Technol.*, vol. 12, no. 4, pp. 1–22, 2021.

[42] Y. Liu, X. Wang, S. Wu, and Z. Xiao, "Independence promoted graph disentangled networks," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2020, vol. 34, no. 4, pp. 4916–4923.

[43] S. Fan, X. Wang, Y. Mo, C. Shi, and J. Tang, "Debiasing graph neural networks via learning disentangled causal substructure," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2022, pp. 24934–24946.

[44] Y. Liu, L. Yu, S. Zhao, X. Wang, L. Geng, Z. Xiao, S. Ma, and Y. Pang, "Graph neural networks with adaptive confidence discrimination," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Sep. 4, 2024, doi: 10.1109/TNNLS.2024.3446229.

[45] J. Bo, H. Zhu, and Z. Tang, "Beyond homophily in graph neural networks: A survey and perspective," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 6, pp. 4963–4979, Jun. 2023.

[46] R. Ying, R. He, K. Chen, P. Eksombatchai, and J. Leskovec, "Graph convolutional neural networks on billion-scale graphs," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2021, pp. 1457–1466.

[47] L. He, L. Bai, X. Yang, H. Du, and J. Liang, "High-order graph attention network," *Inf. Sci.*, vol. 630, pp. 222–234, Jun. 2023.

[48] J. Gasteiger, S. Weißenberger, and S. Günnemann, "Diffusion improves graph learning," in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst.*, vol. 32, Jan. 2019, pp. 13366–13378.

[49] H. Pei, B. Wei, K. C. Chang, Y. Lei, and B. Yang, "Geom-GCN: Geometric graph convolutional networks," in *Proc. 8th Int. Conf. Learn. Represent.*, Feb. 2020, pp. 10247–10258.
[50] R. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec, "Graph transformer networks," in *Proc. 34th Adv. Neural Inf. Process. Syst.*, vol. 33, 2021, pp. 11983–11993.
[51] A. Sinha, F. Ribeiro, and J. Leskovec, "Node propagation modeling for improving long-range dependencies in GNNs," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 5, pp. 1021–1033, May 2023.
[52] H. Zhou, Y. Zhang, Z. Huang, and W. Li, "Bridging heterogeneous nodes in graph learning through auxiliary connections," *Inf. Sci.*, vol. 634, pp. 50–65, Oct. 2023.
[53] R. Wang, T. Xu, and F. Wang, "Learning robust graph representations with auxiliary node generation," *Expert Syst. Appl.*, vol. 241, Nov. 2024, Art. no. 122685.
[54] Z. Li, X. Guo, Y. Liu, and J. Wu, "A survey on graph data augmentation techniques for deep learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 3, pp. 1205–1223, Mar. 2023.
[55] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. 37th Int. Conf. Mach. Learn.*, Jan. 2020, pp. 1597–1607.
[56] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, "Graph contrastive learning with augmentations," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2020, pp. 5812–5823.
[57] A. Bondy and U. Murty, *Graph Theory With Applications*. New York, NY, USA: Macmillan, 1976.
[58] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2016, pp. 3844–3852.
[59] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Appl. Comput. Harmon. Anal.*, vol. 30, no. 2, pp. 129–150, Mar. 2011.
[60] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021.
[61] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. 31st Adv. Neural Inf. Process. Syst.*, Jan. 2017, pp. 1024–1034.
[62] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, Apr. 2011, pp. 315–323.
[63] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn.*, Jun. 2010, pp. 807–814.
[64] Q. Guo, X. Yang, M. Li, and Y. Qian, "Collaborative graph neural networks for augmented graphs: A local-to-global perspective," *Pattern Recognit.*, vol. 158, Feb. 2025, Art. no. 111020.
[65] Q. Guo, X. Yang, W. Guan, K. Ma, and Y. Qian, "Robust graph mutual-assistance convolutional networks for semi-supervised node classification tasks," *Inf. Sci.*, vol. 694, Mar. 2025, Art. no. 121708.
[66] W. Zhao, G. Xu, Z. Cui, S. Luo, C. Long, and T. Zhang, "Deep graph structural infomax," in *Proc. AAAI Conf. Artif. Intell.*, Jun. 2023, vol. 37, no. 4, pp. 4920–4928.
[67] M. Steinbach, G. Karypis, and V. Kumar, "A comparison of document clustering techniques," in *Proc. Workshop Text Mining*, May 2000, pp. 428–439.
[68] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad, "Collective classification in network data," in *Proc. Eur. Conf. Mach. Learn.*, vol. 29, Sep. 2008, pp. 93–106.
[69] J. Yang, W. W. Cohen, and R. Salakhutdinov, "A comparative analysis of feature learning in graph-based models for network data," in *Proc. 2016 Int. Conf. Mach. Learn.*, 2016, pp. 1–18.
[70] L. V. D. Maaten and G. E. Hinton, "Visualizing data using t-SNE," in *Proc. J. Mach. Learn. Res.*, vol. 9, Jan. 2008, pp. 2579–2605.
[71] Y. Zheng, L. Yi, and Z. Wei, "A survey of dynamic graph neural networks," *Frontiers Comput. Sci.*, vol. 19, no. 6, Jun. 2025, Art. no. 196323, doi: 10.1007/s11704-024-3853-2.
[72] S. Lundberg and S. Lee, "A unified approach to interpreting model predictions," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, Jan. 2017, pp. 1–10.
[73] M. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you? Explaining the predictions of any classifier," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Demonstration*, 2016, pp. 1135–1144.

XIANGYI WANG was born in Zhumadian, Henan, China, in 2005. He is currently pursuing the bachelor's degree with Jiangsu University of Science and Technology. His main research interests include deep learning applications and computer vision.

FENGJUN ZHANG received the Ph.D. degree from Nanjing University of Aeronautics and Astronautics, in 2022. His research interests include machine learning, medical imaging, and intelligent hardware.

WEI TENG received the M.E. degree from the School of Computer, Jiangsu University of Science and Technology, in 2004. She is currently a Lecturer. Her research interests include machine learning, network security, and privilege escalation.

BAODA LIU received the B.S. degree from Southwest University of Science and Technology, China, and the M.E. degree from Yangzhou University. He is currently a Lecturer with the School of Computer, Jiangsu University of Science and Technology. His research interest includes ideological and political education of new quality productive forces.