L2DGCN: Learnable Enhancement and Label Selection Dynamic Graph Convolutional Networks for Mitigating Degree Bias

Jingxiao Zhang[1], Shifei Ding[1,2,*], Jian Zhang[1], Lili Guo[1], Xuan Li[1]
[1] School of Computer Science and Technology/School of Artificial Intelligence,
China University of Mining and Technology, Xuzhou 221116, China
[2]Mine Digitization Engineering Research Center of Ministry of Education,
China University of Mining and Technology, Xuzhou 221116, China
{zhangjingxiao, dingsf, zhangjian10231209, liliguo, lixuan23}@cumt.edu.cn

Abstract

Graph Neural Networks (GNNs) are powerful models for node classification, but their performance is heavily reliant on manually labeled data, which is often costly and results in insufficient labeling. Recent studies have shown that message-passing neural networks struggle to propagate information in low-degree nodes, negatively affecting overall performance. To address the information bias caused by degree imbalance, we propose a **L**earnable **E**nhancement and **L**abel **S**election **D**ynamic **G**raph **C**onvolutional **N**etwork (**L2DGCN**). L2DGCN consists of a teacher model and a student model. The teacher model employs an improved label propagation mechanism that enables remote label information dissemination among all nodes. The student model introduces a dynamically learnable graph enhancement strategy, perturbing edges to facilitate information exchange among low-degree nodes. This approach maintains the global graph structure while learning graph representations. Additionally, we have designed a label selector to mitigate the impact of unreliable pseudo-labels on model learning. To validate the effectiveness of our proposed model with limited labeled data, we conducted comprehensive evaluations of semi-supervised node classification across various scenarios with a limited number of annotated nodes. Experimental results demonstrate that our data enhancement model significantly contributes to node classification tasks under sparse labeling conditions.

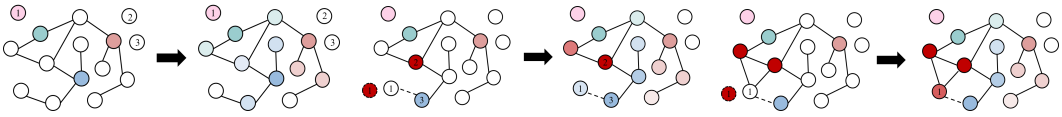
1 Introduction

Graphs model structured and relational systems and are used in fields like traffic networks[1], molecular structures[2], and protein networks[3]. Graph learning algorithms analyze graph-structured data by considering node features and their relationships (edges), achieving success in many domains.

Graph neural networks (GNNs), based on message-passing mechanisms, are a key technology for handling graph data. Node classification, a core task related to graphs, has received much attention. Traditional GNN methods rely on supervised learning with numerous labeled nodes, but high labeling costs limit their practicality. Researchers have combined self-training and pseudo-labeling techniques with GNNs to improve semi-supervised node classification under limited labeled data. However, these methods still struggle when labeled nodes are scarce or unlabeled nodes are abundant.

*Corresponding author

Preprint. Under review.



(a) (b) (c)

Figure 1: The number of nodes corresponding to the node degree.

In a graph, a node's degree usually refers to the number of edges connected to that node, reflecting its connection strength and importance within the network, significantly influencing the path and speed of information propagation. However, in real-world scenarios, the degree distribution of nodes often follows a power-law distribution[4], where most nodes are low-degree nodes, and only a few have very high degrees. We believe that this degree imbalance is a primary reason for the model's failure in situations with few labeled nodes. According to related influence theories[5], the label of a target node is affected by the cumulative influence of the normalized features of its neighboring nodes within a K-hop propagation range. For low-degree nodes, every connection is crucial for their correct classification, but due to their limited number of connections, their information propagation capacity is significantly restricted, thus affecting the effectiveness of the propagation path. Figure 1 illustrates the impact of the degree of the node on the propagation of information when the number of labeled nodes is limited. In Figure 1 (a), the degree-0 node (labeled node 1 and unlabeled nodes 2 and 3) cannot transmit information; in Figure 1 (b), the 1-degree node is misclassified due to incorrect connections; whereas in Figure 1 (c), despite the presence of some erroneous connections, nodes with degrees greater than 1 still achieve correct classification. The insufficient connections of low-degree nodes limit information propagation, especially in the case of scarce labeled nodes. Therefore, in this context, graph neural networks require multi-step propagation to effectively transmit information, highlighting the importance of addressing low-degree nodes in the effective dissemination of information.

In the research on the degree bias problem in Graph Neural Networks (GNNs), several studies have explored solutions from different perspectives: DegFairGT [6] proposes a learnable structural enhancement and structural self-attention mechanism, which generates new edges by calculating the structural similarity between node pairs to balance message passing, and retains the global topology with the help of a self-supervised task based on p-step transition probability matrices, thereby alleviating the problem of insufficient information in low-degree nodes and over-smoothing in high-degree nodes; DAHGN [7], on the other hand, focuses on heterogeneous information networks, constructs a dual-view contrast framework of heterogeneous views and homogeneous subgraphs, combines semi-supervised task loss and contrast loss, and adopts differentiated strategies for low-degree and high-degree nodes to eliminate degree bias, filling the gap in research on heterogeneous scenarios; GraphPatcher [8] innovatively realizes model-agnostic degree bias mitigation through test-time augmentation, iteratively generates virtual nodes to repair damaged neighborhoods of low-degree nodes, improving the performance of low-degree nodes (by an average of 6.5%) while preserving the advantages of high-degree nodes, with an overall performance improvement of 3.6% on average. However, existing methods still have limitations in sparse labeling scenarios: DegFairGT's structural enhancement has limited adaptability to dynamic topologies and is difficult to directly cope with the challenge of insufficient labeled data; DAHGN's contrastive learning framework is insufficient in handling the reliability of pseudo-labels in complex heterogeneous networks; GraphPatcher, as a test-phase strategy, cannot fundamentally optimize the information aggregation of low-degree nodes during training.

Our proposed learnable enhanced and label selection dynamic graph convolutional network aims to address the poor node classification performance caused by degree bias in the absence of labeled nodes. The model comprises a teacher model and a student model. The teacher model generates pseudo-labels for unlabeled nodes via an improved label propagation method. The student model dynamically learns from the graph through two approaches: 1) Structural Optimization: It prunes edges based on node degree, retains core edges, and uses high-order feature information from a decoupled GCN to enhance the topological structure of low-degree nodes, ensuring balanced information flow. 2) Pseudo-Label Selection: A pseudo-label selector combines nodes with high confidence and removes those with low confidence, dynamically updating the training set. Our contributions include:

- Proposing a dynamic graph convolutional network based on learnable augmentation and label selectors to address degree bias caused by the scarcity of labeled nodes.
- Introducing a teacher model that uses soft pseudo-label propagation to expand the training set.
- Designing a student model that performs edge pruning based on node degree and integrates higher-order node features for dynamic topology learning, thereby mitigating degree bias effects while using the label selector to enhance the training set.
- Demonstrating through extensive experiments on multiple datasets that our model is highly effective for semi-supervised node classification with extremely limited labeled nodes, particularly in alleviating degree bias.

2 Related Work

Graph Convolutional Networks Graph Convolutional Networks (GCNs) have significantly advanced graph learning, being mainly divided into spectral and spatial graph convolutions. Spectral graph convolution processes graph signals using graph spectral theory, Fourier transforms, and convolution theorems[9][10], while spatial graph convolution extracts features by passing and aggregating information from neighboring nodes. Common GCN models such as GCN [11], GAT [12], and SGC [13] typically employ a coupled structure of propagation and aggregation. However, in scenarios with scarce labeled nodes, increasing model depth is often considered a solution, though deeper models like GCNII[14] see a significant increase in computational complexity. Consequently, decoupled GCNs have gained attention for their stability and flexibility. For instance, some approaches expand the receptive field through decoupled transformation and propagation, APPNP[15] optimizes global information utilization by combining personalized PageRank, GAMLP [16] maintains high scalability during pre-computation, while DecGCN [17] focuses on improving model stability and generalization capabilities.

Node Classification With Few Labels In semi-supervised node classification tasks with limited labeled nodes, traditional GCN models often face performance issues due to insufficient supervision. Recent graph learning methods aim to address this challenge. For example, CGPN [18] uses Poisson learning to counteract Laplacian performance degradation, M3S [19] enhances GCN generalization through a self-supervised multi-stage training framework, IGCN [20] introduces a unified graph filtering approach to reduce overfitting and training parameters, GraphHop [21] improves graph signal smoothing with a two-stage training process, AGST [22] enhances decision boundary separation by capturing remote node interactions through self-training, and CMPGNN [23] presents a noise-resistant framework via contrastive message passing. PASTEL [24] addresses the problems of insufficient information and excessive suppression caused by "topological imbalance" by proposing a position-aware graph structure learning framework. It enhances intra-class connections and optimizes edge weights through anchor position encoding, alleviating structural biases at the level of propagation paths. NodeMixup [25] focuses on the insufficient reachability between labeled and unlabeled nodes, designing a cross-set mixing strategy and neighbor label distribution-aware sampling to enhance information interaction through node pair mixing without adjusting the GNN architecture. Another study[26] expands the GNN's receptive field to skip neighborhoods through position encoding, adding virtual nodes/edges to the input graph and injecting position features to achieve model-agnostic receptive field expansion, avoiding complex architectural modifications. Despite these advances, existing methods often overlook the reliability of pseudo-labels, which can lead to inaccuracies in model training, MSP-LR[27] introduces a label regularization method and proposes a graph neural network with basic learning and label regularization modules to enhance label reliability through pseudolabeling and regularization based on the cluster assumption.

Graph Representation Learning In graph representation learning, edges are crucial for information dissemination, but real-world graphs often contain noisy edges. Researchers have developed several approaches to optimize graph structures. JLGCN [28] transforms graph optimization into distance metric learning using the Mahalanobis distance metric. [29] proposes an end-to-end joint fusion framework aiming for consistent feature integration and adaptive topology tuning. IDGL [30] frames graph learning as similarity metric learning, iteratively refining graph structures and embeddings. [31] introduces a self-supervised framework that combines graph structure learning, clustering for pseudo-labels, and sample selection for clean labels. [32] leverages information theory to maximize mutual information for reconstructing topological transformations. [33] presents a

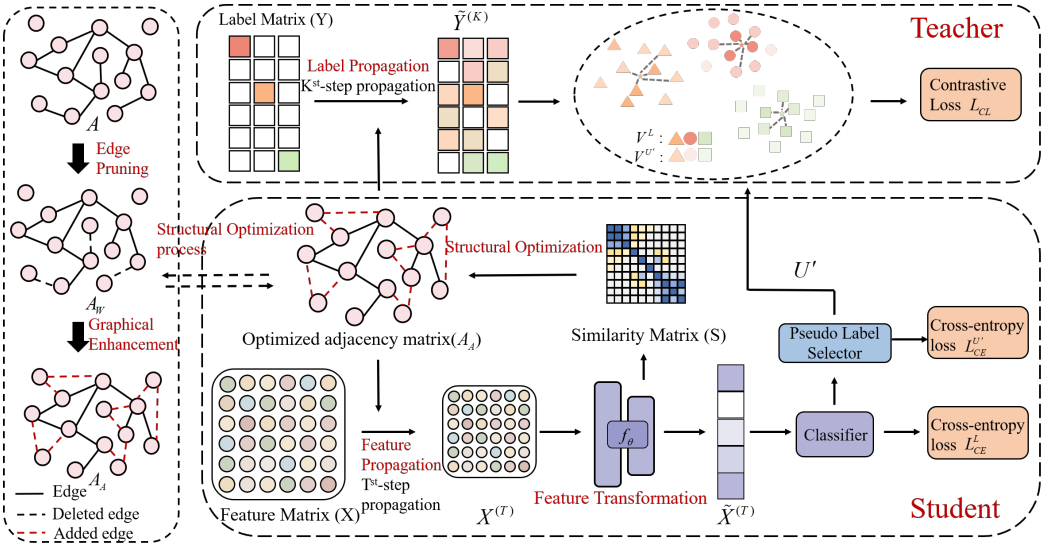


Figure 2: Overview of the proposed framework. In each iteration, the teacher model generates pseudo-labels for all nodes, while the student model enhances the graph structure using higher-order features. Based on the student model's pseudo-labels, reliable nodes are selected to dynamically update the training set and optimize pseudo-labeling via weakly supervised contrastive loss.

method that simultaneously integrates graph learning and graph convolution into a unified network architecture and enforces label smoothing through unsupervised loss terms.[34] uses graph structure refinement to eliminate irrelevant noise and simultaneously maximizes view-shared and view-unique task-relevant information, thereby tackling the frontier of non-redundant multiplex graph.

3 Proposed Method

This section describes our proposed node classification model for under-labeled scenarios. We start with a mathematical formulation of the problem, followed by an overview of the model's architectural framework and a detailed exposition of its key components in the subsequent subsections.

3.1 Problem formulation

An undirected graph with e edges and n nodes can be represented as a quintuple:

$$G = (V, E, X) \tag{1}$$

where V and E denote the set of nodes and edges, respectively, and $X \in \mathbb{R}^{n \times d}$ denotes the initial feature matrix of the graph G with n nodes each having d features. The adjacency matrix $A \in \{0,1\}^{n \times n}$ represents the connections between nodes, and $\tilde{A}$ includes self-loops. Let $\tilde{D}$ be the diagonal matrix of $\tilde{A}$, and $\hat{S}_{\text{sym}} = \tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}$ denotes the symmetric normalized adjacency matrix with self-loops. The labeling matrix is denoted by $Y \in \mathbb{R}^{n \times c}$, where c is the number of classes. In a semi-supervised setup, the node set V is divided into labeled V^L and unlabeled V^U sets. Our focus is on predicting labels for V^U given a few labeled nodes in V^L, which may be balanced or unbalanced across classes. If each class has K labeled nodes, the problem becomes under-shot semi-supervised node classification.

3.2 Architecture Overview

In this section, we will provide a detailed introduction to our proposed L2DGCN, a self-training graph convolutional network model. This model effectively improves the performance of GCN when labeled nodes are scarce by mitigating degree bias through three innovative designs across two key modules. Figure 2 illustrates the overall architecture of L2DGCN.

3.2.1 Teacher model based on the propagation of soft pseudo-labels

In self-training models, teacher models typically use labelled data to predict unlabelled data, providing pseudo-labelling information for unlabelled nodes.Whereas in the label propagation algorithm [35], the labels of the tagged nodes are reset to their true labels after each iteration, which means that the influence of other nodes on the labels of the tagged nodes is absorbed in the subsequent iterations.Therefore, we make use of soft pseudo labels for long range propagation in order to generate pseudo labels for unlabelled nodes. The improved soft label propagation formula is as follows:

$$\begin{aligned} \tilde{Y}^{(K)} &= (1-\alpha)\hat{S}_{\text{sym}}^{\text{iter.}}\tilde{Y}^{(K-1)} + \alpha Y^{(0)} \\ \hat{S}_{\text{sym}}^{\text{iter.}} &= (\tilde{D}_A^{\text{iter.}})^{-1/2}\tilde{A}_A^{\text{iter.}}(\tilde{D}_A^{\text{iter.}})^{-1/2} \end{aligned} \tag{2}$$

where $Y^{(0)} = Y \in R^{n*c}$is the initial label matrix, K is the order of propagation, $\tilde{A}_A^{iter.} = A_A^{iter.} + I$ denotes the adjacency matrix with self-loop added after the iter.th matrix optimization, and $\tilde{D}_A^{iter.}$ is the diagonal matrix of $\tilde{A}_A^{iter.}$, where $\tilde{d}_i^{iter.} = \sum_{j=1}^{n} \tilde{A}_{ij}^{iter.}$, $D_{ii}^{iter.} = d_i^{iter.}$,$\hat{S}_{sym}^{iter.}$ is the normalized adjacency matrices, and for the convenience of writing, in the following expressions we omit the superscript iter. α is the balanced higher-order pseudo-labels with the initial pseudo-labels hyperparameters. By setting an appropriate α , the model can effectively maintain accurate perception of local structures even after multiple propagations, thus improving the overall learning effect and model performance.

3.2.2 Student model based on dynamic learnable graph augmentation and label selector

Decoupled GCN backbone

Graph Convolutional Networks (GCNs) combine neighbourhood aggregation and feature transformation for node representation learning. However, recent studies indicate that this coupled design can cause issues like training difficulties, underutilisation of graph structures, and excessive smoothing. To address these, we adopt a decoupled GCN model for graph feature propagation, enabling higher-order feature interactions.

$$\begin{aligned} X^{(T)} &= (1-\beta)\hat{S}_{\text{sym}}X^{(T-1)} + \beta X^{(0)} \\ \tilde{X}^{(T)} &= (\text{ReLU}(X^{(T)}W_1))W_2 \end{aligned} \qquad T \geq 1 \tag{3}$$

where β is a positive parameter that balances the initial and higher-order feature information, similar to α. By setting β appropriately, the model can preserve initial feature information even with infinite propagation. At the first iteration, $X^{(0)} = X$ represents the initial feature information. While a large number of propagation steps T allows for extensive higher-order interaction, excessive T introduces noise, making the classification boundaries less distinct.

Dynamic Learnable Graph Enhancement

Real-world graphs have many low-degree nodes whose edges are crucial for information dissemination. Incorrect or missing edges can significantly affect these nodes. To boost model performance, especially with limited labeled data, we dynamically optimize the graph topology using node degrees and features.

Edge Pruning Based on Node Degrees A node's degree is its edge count. High-degree nodes are central to the graph's structure and information flow, while edges of low-degree nodes are especially important. We propose pruning the graph by keeping edges of high-degree nodes and removing those of low-degree nodes to reduce the impact of incorrect edges. The steps are as follows.

Step 1: Determine the number of nodes to preserve the edges:

$$M = (1-\beta_w) * n \tag{4}$$

where β_w is the pruning rate $\beta_w \in [0,1]$,$(1-\beta_w)$ is the ratio of the number of total nodes to the number of nodes whose edges are to be retained, n is the total number of nodes, and M is the number of nodes to be retained.When $\beta_w = 1$ our model can degenerate into an MLP with self-training,when $\beta_w = 0$ our model can be seen as an APPNP model with self-training.

Step 2: Sort the nodes in descending order based on the node degree d(v) and select the first M nodes, denoted as V_s:

$$V_s = \{v_1, v_2, \ldots, v_M\} \tag{5}$$

where $d(v_1)d(v_2)\dots d(v_M)$, D is the degree matrix , $A \in R^{n\times n}$ is the initial adjacency matrix, where the choice of optimising the initial graph structure each time avoids introducing cumulative errors and local optimal solution problems in the optimisation process, and ensures the controllability and stability of the optimisation process.

Step 3: For the selected node V_s, all edges E_s connected to it are retained, while all edges of the nodes that are not selected and those that are also not selected when the retained edges are selected are removed. The obtained selected edges and the pruned adjacency matrix A_W are

$$E_{_s} = E_s \cup e_{_{ij}},\ \text{i or j } \in V_s \tag{6}$$

$$A_W(i,j) = \begin{cases} 1,\ \text{e}_{\text{ij}} \in E_s \\ 0,\ \text{otherwise,} \end{cases} \tag{7}$$

This operation filters the top M nodes by degree and retains the edges between them. By selecting the right q value, the model can retain the minimal edges for optimal results, offering insights into graph data compression and preservation.

Graph enhancement based on dynamic similarity matrix The decoupled GCN's backbone supports long-range feature propagation and local attention, enabling the feature matrix to capture global and local features. The similarity matrix from this feature matrix effectively expresses both types of features and is calculated as:

$$S(\tilde{x}_i^{(T)}, \tilde{x}_j^{(T)}) = \frac{(\tilde{x}_i^{(T)})^{\text{Trans}} \tilde{x}_j^{(T)}}{\|\tilde{x}_i^{(T)}\| \|\tilde{x}_j^{(T)}\|} \tag{8}$$

where $S(\tilde{x}_i^{(T)}, \tilde{x}_j^{(T)})$ denotes the similarity between nodes i and j.

To refine the graph structure and ensure low-degree nodes primarily have core edges, we use a KNN graph for unselected discrete nodes V_d, guaranteeing each has at least P neighbors. The graph structure is enhanced by:

$$A_A(i,j) = \begin{cases} 1, & \text{if } e_{ij} \in A_w \text{ or } S(\tilde{x}_i^{(T)}, \tilde{x}_j^{(T)}) \geq \min(\tau(\tilde{x}_i^{(T)}, P), \tau(\tilde{x}_j^{(T)}, P)), \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

Here, $e_{ij} \in A_w$ indicates A_W contains edge e_{ij}, and $\tau(\tilde{x}_i^{(T)}, P)$ returns the similarity between $\tilde{x}_i^{(T)}$ and the P-th similar row vector in $\tilde{X}^{(T)}$. This approach leverages the mutual reinforcement between quality features and good structural information to optimize the graph structure for subsequent model iterations.

Self-training label enhancement Self-training augments GNNs by generating pseudo-labels for unlabeled nodes using a teacher model, but initial pseudo-labels can be unreliable. We introduce a method with a confidence threshold μ, where only pseudo-labels above μ are used, ensuring high-confidence contributions. The student model determines node confidence via the formula:

$$\begin{aligned} \widetilde{Y}^{(0)} &= (\text{ReLU}(XW_1))W_2 \\ \widetilde{Y}^{(T)} &= (1-\beta)\hat{S}_{\text{sym}}\widetilde{Y}^{(T-1)} + \beta\widetilde{Y}^{(0)} \end{aligned} \qquad T \geq 1 \tag{10}$$

In the student model, parameter T controls information dissemination. A larger T captures broader node info, aiding complex graph understanding but risking noise overload in suboptimal graphs. A smaller T limits info to local nodes, possibly missing higher-order relationships. Balancing info coverage and noise is key for effective learning.

Next, according to the set threshold μ, the high confidence unlabelled nodes are selected for supervised learning of the model, i.e:

$$V^{U'} = V^{U'} \cup \{v_i, \tilde{y}_i^{(T)}\} \quad \text{for } i \in V^U \text{ such that } \tilde{y}_i^{(T)} \geq \mu \tag{11}$$

where V^U is the initial set of unlabelled nodes and $V^{U'}$ is the set of reliable unlabelled nodes.

In order to utilise reliable pseudo-labels to assist in labelling the training set for model training, the standard semi-supervised learning objective function can be modified into the following form:

$$L = L_{ce}^{L} + \lambda L_{ce}^{U'} \tag{12}$$

where L_{ce}^{L} is the cross-entropy loss of the labelled nodes and $L_{ce}^{U'}$ is the cross-entropy loss of the unlabelled nodes in the student model with a confidence level greater than μ. i.e:

$$L_{ce}^{L} = -\sum_{v_i \in V^L} \sum_{c=1}^{C} y_i^c \log \tilde{y}_i^c \tag{13}$$

$$L_{ce}^{U'} = -\sum_{v_i \in V^{U'}} \sum_{c=1}^{C} \hat{y}_i^c \log \tilde{y}_i^c \tag{14}$$

where $\hat{y}_i^c$ denotes the pseudo-labelling information obtained by the teacher model and $\tilde{y}_i^c$ denotes the pseudo-labelling obtained through the student model.Also, in order to maintain the consistency of the information obtained from the teacher model and the student model, we motivate the similarity function to assign large values to the positive pairs and small values to the negative pairs by introducing the infoNCE contrast loss [36, 37].

$$L_{InfoNCE} = \sum_{i=1}^{n} -\log \frac{\exp(z_i \cdot z_i^+ / \tau)}{\sum_{i=0}^{r} \exp(z_i \cdot z_j / \tau)} \tag{15}$$

Let z_i^+ be the positive sample of z_i, and z_j consist of one positive and r negative embeddings. τ adjusts the model's ability to distinguish negative samples; a large τ may equalize negative sample treatment, while a small τ can hinder convergence or generalization. The contrast loss aims to enhance pseudo-label validity by calculating the distance between reliable pseudo-labeled nodes U' and class prototypes. The formula for class prototypes is:

$$C_c = \frac{1}{|V_c^L + V_c^{U'}|} \sum_{v_i \in \{V_c^L \cup V_c^{U'}\}} \tilde{x}^T \tag{16}$$

where V_c^L denotes the set of labelled nodes belonging to class c, $V_c^{U'}$ denotes the set of reliable pseudo-labelled nodes belonging to class c, and C_c denotes the corresponding class prototype of class c. The hard pseudo-tag of the reliable pseudo-tag set U' is $\hat{y}_i = \arg\max_j \hat{y}_i^j$. The contrast loss function after correction using reliable pseudo-tags is:

$$L_{CL} = \sum_{v_i \in \{V_c^L \cup V_c^{U'}\}} \left(-\log \frac{\exp(x_i \cdot C_{\hat{y}_i} / \tau)}{\sum_{c=1}^{C} \exp(x_i \cdot C_c / \tau)} \right) \tag{17}$$

where $C_{\hat{y}_i}$ is the corresponding prototype of node v_i. For any node p, its corresponding class prototype is used as a positive sample, and the embedding of other class prototypes is a negative sample. Then the loss function of the whole model is:

$$L = L_{ce}^{L} + \lambda L_{ce}^{U'} + \gamma L_{CL} \tag{18}$$

where λ and γ are hyperparameters that balance the loss of unlabelled nodes and contrast.

3.3 Computational Complexity Analysis

The computational complexity of the proposed model is determined by the core operations of both the teacher model (for pseudo-label generation) and the student model (for dynamic graph enhancement, feature propagation, and pseudo-label selection), depending on key parameters: graph scale (number of nodes n, number of edges E), feature dimension d, and propagation steps K (for the teacher model) or T (for the student model); to align the labels (including pseudo-labels) generated by the teacher model with the node pseudo-labels obtained by the student model and ensure consistency in their iterative propagation processes, we set $K = T$. Specifically, the teacher model generates pseudo-labels via improved soft label propagation, with a time complexity of $\mathcal{O}(K \cdot n^2 c)$ for K iterations (where c denotes the number of classes), as each iteration involves matrix multiplication between an $n \times n$ adjacency-related matrix and an $n \times c$ label matrix. For the student model, its complexity comes from three modules: dynamic graph enhancement (with complexity $\mathcal{O}(n \log n + E + n^2 d + n \cdot P \cdot d)$, including node sorting, edge traversal, KNN graph construction

based on cosine similarity calculation, and neighbor supplementation, where P is the number of neighbors per node), decoupled GCN propagation (with complexity $\mathcal{O}(T \cdot E \cdot d)$ for T iterations, as each iteration requires feature aggregation over E edges for d-dimensional features), and pseudo-label selection (with complexity $\mathcal{O}(n \cdot c)$ from traversing all n nodes to compute label confidence). Combining the teacher and student modules, the overall computational complexity is dominated by terms related to n (notably n^2 from matrix operations), E (notably $E \cdot d$ from graph propagation), and the unified propagation step $K = T$.

4 Experiments

In this section, we conduct experiments to validate our model's effectiveness and robustness with extremely few labeled nodes. Experimental design is as follows:

4.1 Experimental setup

Datasets We validated our model's effectiveness in semi-supervised node classification on six homogeneous graph datasets of varying sizes. These include citation networks Cora, Citeseer, and Pubmed[38], widely used for semi-supervised node classification; Coauthor-CS and Coauthor-Physics[39] datasets for academic collaboration analysis; and Amazon-Photo[39], which comprises product images with metadata labels. Detailed dataset information is in Table 1.

Table 1: Dataset statistics

Dataset	#Nodes	#Edges	#Features	#Classes	#Edge Density
Cora	2,708	5,278	1,433	7	0.0014
Citeseer	3,327	4,552	3,703	6	0.0008
Pubmed	19,717	44,324	500	3	0.0002
Coauthor-CS	18,333	81,894	6,805	15	0.0005
Coauthor-Physics	34,493	247,962	8,415	5	0.0004
Amazon-Photo	7,487	119,043	745	8	0.0042

Table 2: Hyperparameters range

Hyperparameters	#Range
$iter.$	$\{1, 2, 3, 4, 5\}$
β_w	$\{0.12, 0.10, 0.08, 0.06, 0.04, 0.02, 0.00\}$
μ	$\{1.0, 1.2, 1.4, 1.6, 1.8, 2.0\}$
P	$\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12\}$
T	$\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$

Balancing setup at low labelling rates: We built the training set by randomly selecting 3, 5, or 10 labeled nodes per category (3-shot, 5-shot, 10-shot). The validation set had 30 nodes per category, with the rest used for testing [22].

Balancing setup under standard segmentation: Following [11], we used 20 labeled nodes per class for training, with a 500-node validation set and the remaining nodes in the test set.

Compared Methods: We compared our model with classical and state-of-the-art methods, including GCN [11], GAT [12], SGC [13], label-efficient GCN models (GLP [40], IGCN [40], CGPN [41], CMPGNN [23], GraphHop [21]), and self-trained GNN models (PTA [15], ST-GCNs [42], M3S [19], AGST [22], Muse [43]).

4.2 Implementation Details

We implemented all algorithms in PyTorch with the Adam optimizer, following original settings when available. Results reported are average accuracies from 10 independent runs. Our model was trained with a maximum of 100 iterations, a learning rate of 0.01, and a regularization weight of 5×10^{-5}. To prevent overfitting, early stopping was applied, halting training if validation loss didn't improve for 1000 steps. Default parameters included K=T, matrix optimization iterations (iter.), pruning rate β_w, P neighboring nodes added, and T feature propagation steps for the decoupled GCN. Hyperparameter ranges are in Table 2, with variations based on dataset labeling rates.For the selection of optimal values of the parameters listed in 2, we adopted the grid search method. Specifically, within the preset parameter ranges, we exhaustively combined the possible values of each parameter, evaluated the model performance on the validation set, and finally selected the parameter combination that made the model perform optimally.

4.3 Main results

In the experiments, we evaluate our model and baselines in semi-supervised node classification across various labeling rates. The top three models are labeled, with the best in bold and the other two underlined, combining results from [22] and our experiments.

Table 3: Accuracy (%) of semi-supervised node classification test with low labeling rate under balanced training

Method	Cora			Citeseer			Pubmed		
	3-shot	5-shot	10-shot	3-shot	5-shot	10-shot	3-shot	5-shot	10-shot
LP	52.76 ± 0.92	58.72 ± 0.79	64.03 ± 0.65	34.87 ± 0.93	37.58 ± 0.81	41.74 ± 0.50	59.58 ± 0.98	62.32 ± 0.94	67.02 ± 0.75
GCN	56.31 ± 0.81	64.18 ± 0.66	72.87 ± 0.53	47.59 ± 0.90	54.27 ± 0.81	62.26 ± 0.57	59.24 ± 0.81	66.40 ± 0.85	72.37 ± 0.74
GAT	63.39 ± 0.98	69.93 ± 0.84	76.44 ± 0.35	51.62 ± 0.97	58.67 ± 0.81	65.13 ± 0.51	64.72 ± 0.91	68.32 ± 0.90	73.85 ± 0.60
SGC	55.94 ± 0.97	59.77 ± 0.97	67.76 ± 0.91	52.60 ± 0.92	58.94 ± 0.85	64.92 ± 0.54	58.74 ± 0.92	64.72 ± 0.91	69.02 ± 0.83
GLP	65.99 ± 0.94	72.31 ± 0.89	77.56 ± 0.43	50.46 ± 0.96	59.09 ± 0.88	66.06 ± 0.38	66.31 ± 0.95	72.59 ± 0.73	75.82 ± 0.58
IGCN	66.91 ± 0.91	72.78 ± 0.85	78.27 ± 0.31	50.99 ± 0.97	59.53 ± 0.89	66.51 ± 0.39	66.23 ± 0.97	71.96 ± 0.85	75.97 ± 0.50
CGPN	71.88 ± 2.52	71.83 ± 3.14	74.85 ± 1.54	62.54 ± 3.56	62.20 ± 1.63	63.76 ± 1.09	68.21 ± 3.89	71.21 ± 2.90	75.44 ± 2.53
CMPGNN	66.35 ± 2.65	77.48 ± 3.15	77.80 ± 1.27	55.85 ± 1.55	60.50 ± 2.16	64.43 ± 1.93	67.49 ± 2.66	71.75 ± 2.53	72.65 ± 1.86
GraphHop	69.06 ± 2.23	74.60 ± 1.18	75.49 ± 0.93	57.56 ± 0.46	60.39 ± 3.25	64.43 ± 4.71	61.24 ± 3.35	64.11 ± 6.94	73.61 ± 3.08
NAGphormer	63.82 ± 0.87	70.28 ± 0.41	74.88 ± 0.97	48.50 ± 0.98	53.22 ± 0.72	59.26 ± 0.58	67.07 ± 0.53	68.13 ± 0.38	73.49 ± 0.23
PTA	69.21 ± 0.99	73.98 ± 0.73	78.69 ± 0.39	54.18 ± 0.94	61.13 ± 0.86	66.69 ± 0.48	67.69 ± 0.92	72.28 ± 0.82	76.47 ± 0.51
ST-GCNs	65.85 ± 0.94	71.16 ± 0.87	76.54 ± 0.49	49.85 ± 0.95	61.39 ± 0.91	68.58 ± 0.36	65.99 ± 0.93	70.26 ± 0.98	74.10 ± 0.63
M3S	64.01 ± 0.71	69.26 ± 0.75	77.20 ± 0.41	50.31 ± 0.88	59.72 ± 0.82	65.99 ± 0.41	66.01 ± 0.90	72.38 ± 0.85	75.31 ± 0.49
AGST	71.00 ± 0.53	79.72 ± 0.33	79.92 ± 0.58	52.33 ± 0.62	53.12 ± 0.80	66.44 ± 0.29	**76.66** ± 0.63	**78.26** $\pm$ **0.77**	73.36 ± 0.53
OURS	**78.50** $\pm$ **0.21** (↑**6.62**)	**81.67** $\pm$ **0.33** (↑**1.95**)	**82.29** $\pm$ **0.25** (↑**2.37**)	**68.99** $\pm$ **0.11** (↑**6.45**)	**70.48** $\pm$ **0.23** (↑**8.28**)	**71.64** $\pm$ **0.50** (↑**3.06**)	74.77 ± 0.55 (↓**1.89**)	78.07 ± 0.40 (↓**0.19**)	**80.88** $\pm$ **0.66** (↑**4.41**)

Table 4: Accuracy (%) of semi-supervised node classification test with low labeling rate under balanced training

Method	Coauthor-CS			Coauthor-Physics			Amazon-Photo		
	3-shot	5-shot	10-shot	3-shot	5-shot	10-shot	3-shot	5-shot	10-shot
LP	57.77±0.77	62.09±0.60	66.18±0.36	73.46±0.93	76.94±0.61	80.55±0.41	69.24±0.92	73.43±0.72	77.78±0.61
GCN	77.17±0.79	84.09±0.59	89.01±0.98	82.49±0.88	87.50±0.69	90.78±0.38	69.54±0.99	74.42±0.97	80.30±0.78
GAT	79.66±0.75	85.11±0.49	89.34±0.19	86.07±1.16	89.35±0.48	91.64±0.48	70.47±1.19	77.89±1.05	82.39±1.11
SGC	84.93±0.57	88.11±0.35	90.13±0.99	87.55±0.64	87.68±0.39	91.38±0.31	75.05±0.88	78.73±0.69	84.14±0.45
GLP	84.58±0.61	87.36±0.61	91.59±0.15	89.34±0.99	91.52±0.32	93.02±0.20	75.11±1.19	81.99±0.97	85.33±0.38
IGCN	84.26±0.47	86.45±0.33	90.82±0.13	89.82±0.57	91.33±0.29	92.78±0.21	75.36±0.98	82.10±0.89	85.50±0.32
CGPN	88.96±3.37	89.14±3.27	90.37±2.14	90.06±3.48	91.76±2.33	92.56±2.22	83.57±3.24	84.74±2.63	87.78±2.44
CMPGNN	80.22±2.57	84.29±2.21	88.65±0.82	81.33±1.98	83.28±2.15	87.68±2.38	81.59±3.35	85.42±4.15	86.94±3.83
GraphHop	71.15±4.98	84.39±0.82	82.99±3.94	85.05±1.18	86.55±0.30	91.91±0.12	68.42±1.09	84.43±1.12	86.20±1.30
NAGphormer	85.61±0.38	89.08±0.85	90.55±0.49	86.26±0.55	89.91±0.82	92.54±0.24	74.79±0.35	83.58±0.33	87.85±0.19
PTA	86.56±0.46	89.43±0.31	90.72±0.18	88.62±0.60	90.36±0.53	92.15±0.32	77.43±0.89	82.63±0.76	85.51±0.74
ST-GCNs	88.34±0.46	89.68±0.45	91.39±0.14	87.61±0.69	90.23±0.39	91.75±0.21	73.86±1.53	81.93±1.09	85.54±0.67
M3S	84.11±0.46	86.96±0.41	91.08±0.11	89.12±0.55	91.27±0.31	92.93±0.25	74.96±0.97	81.88±0.93	85.42±0.37
AGST	87.14±0.25	91.30±0.53	89.83±0.39	91.64±0.47	90.88±0.53	93.26±0.36	81.63±0.73	81.83±0.89	85.97±0.58
OURS	**91.45** $\pm$ **0.36** (↑**2.49**)	**91.82±0.45** (↑**0.52**)	**92.18±0.68** (↑**0.59**)	**93.39±0.66** (↑**1.75**)	**94.03±0.44** (↑**2.27**)	**94.65±0.88** (↑**1.39**)	**86.10±0.53** (↑**2.53**)	**87.90±0.69** (↑**2.48**)	**88.56±0.59** (↑**0.71**)

Low-labeling rates setting: Tables 3 and 4 show our model outperforms others, achieving up to 8.28% higher accuracy in the 5-shot. This highlights its effectiveness with few labeled nodes. Classical shallow GCNs struggle with limited data, while models using higher-order and pseudo-label info perform better.

Standard Divisions experiments: Table 5 shows our model gains up to 2.39% accuracy over baselines on Cora, Citeseer, and Pubmed datasets, despite the low-labeling focus, underscoring its superior performance.

4.4 Ablation Study

In this section, we evaluated the performance gains of each model component through ablation experiments, including graph structure optimization based on node degree, pseudo-label reliability, and the effectiveness of contrastive loss. Experiments are performed on the Cora and Citeseer datasets under 3-shot, 5-shot, and 10-shot labeling rates.Results in Figure 3 show that in the Cora dataset, label selection significantly improves performance in 3-shot and 5-shot tasks, while learnable graph augmentation is more effective in the 10-shot task. Conversely, this pattern is reversed in the Citeseer dataset, which is attributed to the dataset's unique characteristics. Compared with models retaining contrastive loss , L2DGCN-LC shows significant performance gaps: on Cora, the gap reaches over 20 percentage points in 3-shot scenarios , narrowing to 8 and 5 percentage points in 5-shot and 10-shot scenarios, respectively; on the more complex Citeseer, the gaps are over 20, 14, and 7-8 percentage points in 3-shot, 5-shot, and 10-shot scenarios. These results confirm that contrastive loss helps capture feature associations from limited data, suppresses interference in noisy datasets, and is critical for enhancing model adaptability and classification stability. We will supplement these experiments and analyses to clarify the independent role of contrastive loss, making the conclusions more robust.

4.5 Parameter Analysis

In this section, in order to verify the impact of each hyperparameter on the model performance, we conduct a series of experiments on the Cora and Citeseer datasets under 3 and 5 settings. These experiments are divided into three main parts, and their results are shown in Figure 4:

1)Matrix Optimization Hyperparameters: Evaluated pruning rate (β_w) and number of nearest neighbors (P) on Cora and Citeseen. On Cora, both showed minor fluctuations in accuracy. On Citeseer's 3-shot task, accuracy rose with higher (β_w) and more neighbors, but fluctuated; for 5-shot, no clear trend for (β_w), but P showed a rapid improvement before declining if too large. 2) Pseudo-Labelling Selection Threshold (μ): On 5-shot Cora and Citeseer, performance was stable but in Citeseer's 3-shot task, performance varied significantly and dropped sharply as increased. 3) Decoupled GCN Propagation Steps (T): Cora showed minimal variation. Citeseer had notable fluctuations, especially in the 3-shot task, due to many discrete nodes and random training selection.

Table 5: Test accuracy of node classification

Method	Cora	Citeseer	Pubmed
LP	67.04 ± 0.41	45.29 ± 0.34	69.78 ± 0.54
GCN	77.85 ± 0.33	65.95 ± 0.42	76.33 ± 0.47
GAT	76.85 ± 0.34	65.12 ± 0.72	73.20 ± 0.49
SGC	71.19 ± 0.29	69.20 ± 0.37	72.13 ± 0.66
GLP	79.33 ± 0.27	68.94 ± 0.28	78.49 ± 0.39
IGCN	80.11 ± 0.31	67.89 ± 0.29	$\underline{78.64 \pm 0.39}$
CGPN	74.12 ± 1.54	67.34 ± 1.07	75.81 ± 1.26
CMPGNN	72.54 ± 3.32	60.80 ± 1.99	73.94 ± 2.61
GraphHop	79.16 ± 1.10	67.23 ± 2.40	75.62 ± 3.01
PTA	$\underline{81.54 \pm 0.35}$	69.84 ± 0.25	$\underline{78.66 \pm 0.44}$
ST - GCNs	79.75 ± 0.24	70.26 ± 0.23	78.12 ± 0.30
M3S	78.11 ± 0.39	$\underline{70.42 \pm 0.29}$	77.98 ± 0.29
AGST	$\underline{80.57 \pm 0.12}$	67.60 ± 0.01	76.50 ± 0.35
Muse	78.80 ± 0.50	$\mathbf{73.50 \pm 1.40}$	73.60 ± 3.10
OURS	$\mathbf{83.93 \pm 0.11}$ (↑**2.39**)	$\underline{71.79 \pm 0.20}$ (↓**1.71**)	$\mathbf{80.02 \pm 0.01}$ (↑**1.36**)

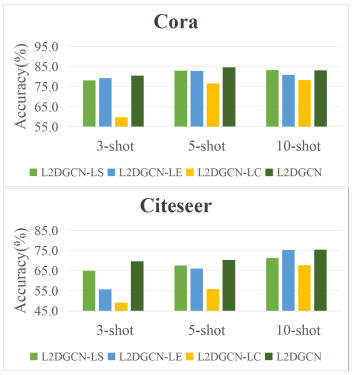


Figure 3: Visualization of ablation results

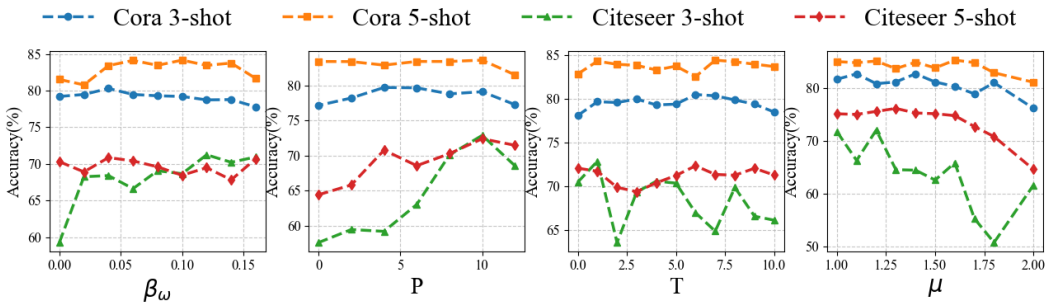


Figure 4: Model performance with varied hyperparameters

5 Conclusion

To address the problem that existing models perform poorly with limited labelled nodes, we propose a new solution, L2DGCN.The model contains two modules and is designed with three key elements aimed at optimising graph structure learning and efficiently propagating semantic information, thus mitigating the problem of model performance degradation due to insufficiently labelled nodes.We have conducted extensive experiments on several benchmark datasets with low labelling rates, and the results show the effectiveness of solving the graph structure problem in terms of node degree, as well as the precise selection of pseudo-labels to address their unreliability.We realise that considering only features for learnable graph augmentation in the current study is not comprehensive enough, especially for low-homogeneous graphs that may introduce undesirable edges.Therefore, future work will focus on a multifaceted exploration of graph augmentation, aiming to develop more robust classification models.

6 Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grants 62276265, 62576344, 62406326, and 62206297.

References

[1] B. Du, X. Hu, L. Sun, J. Liu, Y. Qiao, and W. Lv, "Traffic demand prediction based on dynamic transition convolutional neural network," *IEEE transactions on intelligent transportation systems*, vol. 22, no. 2, pp. 1237–1247, 2021.

[2] J. Jimenez, S. Doerr, G. Martinez-Rosell, A. S. Rose, and G. De Fabritiis, "Deepsite: protein-binding site predictor using 3d-convolutional neural networks," *Bioinformatics*, vol. 33, no. 19, pp. 3036–3042, 2017.

[3] Y. Zhou, Y. Myung, C. H. M. Rodrigues, and D. B. Ascher, "Ddmut-ppi: predicting effects of mutations on protein-protein interactions using graph-based deep learning," *Nucleic acids research*, vol. 52, pp. 207–214, 2024.

[4] X. Tang, H. Yao, Y. Sun, Y. Wang, J. Tang, C. Aggarwal, P. Mitra, and S. Wang, "Investigating and mitigating degree-related biases in graph convolutional networks," in *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*, 2020, pp. 1435–1444.

[5] H. Wang and J. Leskovec, "Combining graph convolutional neural networks and label propagation," *ACM Transactions on Information Systems*, vol. 40, no. 4, pp. 1–27, 2022.

[6] V. T. Hoang, H.-J. Jeon, and O.-J. Lee, "Mitigating degree bias in graph representation learning with learnable structural augmentation and structural self-attention," *IEEE Transactions on Network Science and Engineering*, vol. 12, no. 5, pp. 3656–3670, 2025.

[7] M. Zhao and A. L. Jia, "Dahgn: Degree-aware heterogeneous graph neural network," *Knowledge-Based Systems*, vol. 285, p. 111355, 2024.

[8] M. Ju, T. Zhao, W. Yu, N. Shah, and Y. Ye, "Graphpatcher: mitigating degree bias for graph neural networks via test-time augmentation," in *Proceedings of the 37th International Conference on Neural Information Processing Systems*, 2023.

[9] J. Bruna, W. Zaremba, A. Szlam, and Y. Lecun, "Spectral networks and locally connected networks on graphs," in *2nd International Conference on Learning Representations*, 2014.

[10] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 2016, pp. 3844–3852.

[11] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *5th International Conference on Learning Representations*, 2017.

[12] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *6th International Conference on Learning Representations*, 2018.

[13] F. Wu, A. H. S. Jr., T. Zhang, C. Fifty, T. Yu, and K. Q. Weinberger, "Simplifying graph convolutional networks," in *Proceedings of the 36th International Conference on Machine Learning*, vol. 97, 2019, pp. 6861–6871.

[14] M. Chen, Z. Wei, Z. Huang, B. Ding, and Y. Li, "Simple and deep graph convolutional networks," in *Proceedings of the 37th International Conference on Machine Learning*, vol. 119, 2020, pp. 1725–1735.

[15] H. Dong, J. Chen, F. Feng, X. He, S. Bi, Z. Ding, and P. Cui, "On the equivalence of decoupled graph convolution network and label propagation," in *Proceedings of the Web Conference 2021*, 2021, pp. 3651–3662.

[16] W. Zhang, Z. Yin, Z. Sheng, Y. Li, W. Ouyang, X. Li, Y. Tao, Z. Yang, and B. Cui, "Graph attention multi-layer perceptron," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 4560–4570.

[17] Y. Liu, Y. Gu, Z. Ding, J. Gao, Z. Guo, Y. Bao, and W. Yan, "Decoupled graph convolution network for inferring substitutable and complementary items," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, pp. 2621–2628.

[18] S. Wan, Y. Zhan, L. Liu, B. Yu, S. Pan, and C. Gong, "Contrastive graph poisson networks: Semi-supervised learning with extremely limited labels," in *Advances in Neural Information Processing Systems 34*, 2021, pp. 6316–6327.

[19] K. Sun, Z. Lin, and Z. Zhu, "Multi-stage self-supervised learning for graph convolutional networks on graphs with few labeled nodes," vol. 34, no. 4, 2020, pp. 5892–5899.

[20] Q. Li, X.-M. Wu, H. Liu, X. Zhang, and Z. Guan, "Label efficient semi-supervised learning via graph filtering," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 9574–9583.

[21] T. Xie, B. Wang, and C.-C. J. Kuo, "Graphhop: An enhanced label propagation method for node classification," *IEEE Transactions On Neural Networks and Learning Systems*, vol. 34, no. 11, pp. 9287–9301, 2023.

[22] K. Ding, E. Nouri, G. Zheng, H. Liu, and R. White, "Toward robust graph semi-supervised learning against extreme data scarcity," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 9, pp. 11 661–11 670, 2024.

[23] "Contrastive message passing for robust graph neural networks with sparse labels," *Neural Networks*, vol. 182, p. 106912, 2025.

[24] Q. Sun, J. Li, H. Yuan, X. Fu, H. Peng, C. Ji, Q. Li, and P. S. Yu, "Position-aware structure learning for graph topology-imbalance by relieving under-reaching and over-squashing," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, p. 1848–1857.

[25] W. Lu, Z. Guan, W. Zhao, Y. Yang, and L. Jin, "Nodemixup: tackling under-reaching for graph neural networks," in *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, 2024.

[26] R. Gabrielsson, M. Yurochkin, and J. Solomon, "Rewiring with positional encodings for graph neural networks," *Transactions on Machine Learning Research*, vol. 2023, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:246431122

[27] J. Zhang, S. Ding, J. Zhang, L. Guo, and L. Ding, "Multi-channel set polynomial based label regularized graph neural networks against extreme data scarcity," *Pattern Recognition*, vol. 166, p. 111754, 2025.

[28] J. Tang, W. Hu, X. Gao, and Z. Guo, "Joint learning of graph representation and node features in graph convolutional neural networks," *CoRR*, vol. abs/1909.04931, 2019.

[29] Y. Chen, Z. Wu, Z. Chen, M. Dong, and S. Wang, "Joint learning of feature and topology for multi-view graph convolutional network," *Neural Networks*, vol. 168, pp. 161–170, 2023.

[30] Y. Chen, L. Wu, and M. Zaki, "Iterative deep graph learning for graph neural networks: Better and robust node embeddings," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 19 314–19 326.

[31] J. Yuan, H. Yu, M. Cao, J. Song, J. Xie, and C. Wang, "Self-supervised robust graph neural networks against noisy graphs and noisy labels," *Applied Intelligence*, vol. 53, no. 21, pp. 25 154–25 170, 2023.

[32] X. Gao, W. Hu, and G.-J. Qi, "Self-supervised graph representation learning via topology transformations," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 4, pp. 4202–4215, 2023.

[33] F. Dornaika, J. Bi, and C. Zhang, "A unified deep semi-supervised graph learning scheme based on nodes re-weighting and manifold regularization," *Neural Networks*, vol. 158, pp. 188–196, 2023.

[34] Z. Shen, S. Wang, and Z. Kang, "Beyond redundancy: Information-aware unsupervised multiplex graph structure learning," in *Advances in Neural Information Processing Systems*, vol. 37, 2024, pp. 31 629–31 658.

[35] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Scholkopf, "Learning with local and global consistency," in *Advances in Neural Information Processing Systems*, vol. 16, 2004, pp. 321–328.

[36] K. He, H. Fan, Y. Wu, S. Xie, and R. B. Girshick, "Momentum contrast for unsupervised visual representation learning," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9726–9735.

[37] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *CoRR*, vol. abs/1807.03748, 2018.

[38] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad, "Collective classification in network data," *AI magazine*, vol. 29, no. 3, pp. 93–106, 2008.

[39] O. Shchur, M. Mumme, A. Bojchevski, and S. Günnemann, "Pitfalls of graph neural network evaluation," *CoRR*, vol. abs/1811.05868, 2018.

[40] Q. Li, X.-M. Wu, H. Liu, X. Zhang, and Z. Guan, "Label efficient semi-supervised learning via graph filtering," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9574–9583.

[41] S. Wan, Y. Zhan, L. Liu, B. Yu, S. Pan, and C. Gong, "Contrastive graph poisson networks: Semi-supervised learning with extremely limited labels," in *Advances in Neural Information Processing Systems 34*, 2021, pp. 6316–6327.

[42] Q. Li, Z. Han, and X.-M. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, 2018, pp. 3538–3545.

[43] Z. Wang, Q. Zeng, W. Lin, M. Jiang, and K. C. Tan, "Multiview subgraph neural networks: Self-supervised learning with scarce labeled data," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–14, 2024.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: blue[Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect our contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: blue[Yes]

   Justification: The authors have discussed the limitations of the new method in the Conclusion section.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: gray[NA]

Justification: No theoretical result.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: blue[Yes]

   Justification: Our algorithm is easy to understand, and we have provided detailed explanations of the implementation details.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: orange[No]

Justification: We use the publicly available dataset.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: blue[Yes]

   Justification: We have provided detailed parameter analysis in the experimental section, as well as the basis for setting some parameters to fixed values.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: gray[NA]

   Justification: N.A.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: blue[Yes]

   Justification: We have provided the computational resources.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: blue[Yes]

   Justification: We have carefully read the NeurIPS Code of Ethics and our research conforms with it.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: gray[NA]

    Justification: The research is an improvement of existing classification algorithms and does not address social impacts.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: gray[NA]

    Justification: N.A.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: blue[Yes]

    Justification: We have cited the utilized papers.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: gray[NA]

    Justification: N.A.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: gray[NA]

    Justification: N.A.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: gray[NA]

    Justification: N.A.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: gray[NA]

Justification: N.A.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.