IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 19 April 2024, accepted 25 April 2024, date of publication 29 April 2024, date of current version 13 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3394712

RESEARCH ARTICLE

Pseudo Labeling Collaborative Embedding Representation: A Multi-Channel GCNs Framework for Semi-Supervised Node Classification

MENGNAN PANG[1], XUN WANG[1], JINGJING SONG[1], AND PINGXIN WANG[2]
[1]School of Computer, Jiangsu University of Science and Technology, Zhenjiang, Jiangsu 212100, China
[2]School of Science, Jiangsu University of Science and Technology, Zhenjiang, Jiangsu 212100, China

Corresponding author: Xun Wang (wx_just@126.com)

This work was supported by the National Natural Science Foundation of China under Grant 62006099.

ABSTRACT In recent years, Graph Convolutional Networks (GCNs) have emerged as a crucial methodology for handling graph-structured data, exhibiting superior performance in semi-supervised classification tasks. However, most existing GCNs encounter two main issues in real-world scenarios: 1) graph-structured data may be incomplete, for example, containing outlier nodes and noisy edges, which poses a great challenge for GCNs to extract the relation information for classification tasks; 2) the scarcity of labeled data, often limited to a few-shot scenario, hampers the ability of GCNs to learn comprehensive embedding representations. To cope with these issues, we propose a novel framework called pseudo labeling collaborative multi-channel graph convolutional networks (PCM-GCN). First, considering incomplete graph-structured data, we develop two modules: graph generation module and multi-channel fusion module. The graph generation module is designed to extend the raw data to multiple graphs, which avoids being constrained by the effective expression ability of the raw data. Meanwhile, the multi-channel fusion module integrates embeddings from multiple graphs, capturing the complementarity among multiple channels. Second, to address the problem of sparse labels, we develop a confidence-based pseudo labeling module, appending confident data with pseudo label to the labeled set to enlarge the training set. PCM-GCN leverages pseudo labeling to enhance multi-channel embedding fusion, resulting in rich and comprehensive node embedding representation. Extensive experiments on five benchmark datasets have shown that PCM-GCN surpasses other state-of-the-art methods in semi-supervised node classification tasks.

INDEX TERMS Graph convolutional networks, multi-channel, pseudo labeling, semi-supervised classification learning.

I. INTRODUCTION

The graph-structured data [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], which can characterize entities and their associations, is increasingly popular in real-world scenarios, such as social networks, bank fraud detection, urban planning, and knowledge graphs. Meanwhile, Graph Neural Networks (GNNs) have become a powerful paradigm because of achieving their performance in various complex graph learning tasks, such as node classification [1], [11], [12], [14], [19], [21], [22], [23], [24], [25], link prediction [5], [6], [26], [27], [28], [29] and graph classification [7], [8], [30].

In general, GNNs such as Graph Convolutional Network(GCN) [2], Graph Attention Network(GAT) [1], and Diffusion Improves Graph Learning (GDC) [31] have achieved better performance in semi-supervised node classification tasks. In these models, a message-passing

The associate editor coordinating the review of this manuscript and approving it for publication was Zhengmao Li.

© 2024 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

mechanism is typically employed to transform the initial representation of a node into a compact embedding representation. The GCN extends the convolution operator to graph data, and the main idea is to learn the embedding representation of a node by iteratively aggregating the neighbor features of the node. The receptive field of each node depends on the number of convolutions [32]. Meanwhile, in semi-supervised tasks, learning a rich embedding representation via GCN is generally impacted by the amount of labels.

Although GNNs and their variants have achieved great success in semi-supervised tasks, some baseline methods primarily learn node embedding representation based on a given graph structure (adjacency matrix). Note that because edges (relations) in graph-structured data are defined manually, there are potential information discrepancies between the given graph structure and the expected one [33], which leads to possible limitations in the expressive power of the baseline models. Considering the above issues, recent related works [7], [23], [24], [31], [33], [34], [35] have improved the performance of GNNs in semi-supervised node classification tasks by employing a two-channel (or multi-channel) pattern. For example, multiple adjacency matrices are developed from the multi-order neighbor information of each node, such that the adjacency matrices can directly capture the multi-order connectivity between nodes [23], [34], [35], [36]. AM-GCN [37] constructs a k-nearest neighbor graph based on the node feature matrix and then extracts embeddings from the k-nearest neighbor graph, the raw graph, and their combinations simultaneously. Augmented graphs are generated via perturbed node features, and then fusion learning of augmented graphs and input graph is performed simultaneously, called PA-GCN [24]. However, some issues remain to be addressed: (1) for these models, the perspective for developing the reconstruction graphs is single and simple (e.g. multi-order information or node features); (2) there are very few labeled nodes in many real-world scenario node classification tasks, yet a considerable amount of labeled nodes are required for a multi-channel GCN to train a classifier. Based on this, it is natural to consider exploring the relationship between pseudo labeling [13], [14], [32], [38], [39] and multi-channel embedding representations, which integrates unlabeled node information into the multi-channel training process.

Following the above discussion, in this paper, we propose a Pseudo labeling Collaborate Multi-channel Graph Convolutional Network (PCM-GCN) framework for semi-supervised classification learning. Initially, the graph generation module extracts the underlying graph structure embedded in the raw graph by using perturbation algorithms (e.g., k nearest neighbours and edge modification, etc.). The multi-channel fusion module is developed to input multiple underlying graphs into different GCN encoder. Furthermore, considering the homogeneity of the graphs and the consistency among the multi-channel, we design a consistency loss function to ensure that the embedding representations generated by multiple channels are correlated. Finally, to address the issue of not enough labeled data in real scenarios, we introduce the pseudo labeling module to learn the hidden information in unlabeled data to collaborate the multi-channel embedding representations, and then fuse the multi-channel to obtain the final node embedding representation for downstream classification tasks. In summary, the main contributions of this study are described below:

(1) Propose an end-to-end multi-channel neural network framework, which can effectively perform multi-channel node embedding representation based on pseudo labeling collaborative method.
(2) Exploit the node embedding representation of each channel, thereby providing a way for the reliability of the pseudo-labels, i.e., stable pseudo-labels are generated for the unlabeled nodes by considering the proximity of the nodes in the feature and embedding space.
(3) The proposed framework is leveraged to conduct semi-supervised classification tasks, and achieves superior performance compared with other state-of-the-art graph-based learning algorithms. Furthermore, the framework can be generalized to variants of GCNs without any constraints.

The rest of this paper is arranged as below: Section II involves the related work. Our proposed PCM-GCN framework is showed in Section III. The extensive experimental results and analysis of the proposed PCM-GCN are presented in Section IV. Finally, Section V concludes this work.

II. RELATE WORK

Graph Convolutional Networks based on multi-channel fusion have spawned many representative works due to their excellent performance in semi-supervised classification tasks [7], [10], [33], [34], [37], [40]. One of the most noteworthy aspects is the multi-channel fusion model for augmented graphs. Here they share two common traits. (1) Extend a single channel into multiple channels with graph augmentation. (2) Only the labeled data is considered to be added to the training of the multi-channel fusion model. Thus, we discuss related work to our method from the two perspectives below.

Graph Augmentation: Graph augmentation [41], [42] is a data augmentation technique that aims to increase the diversity of training data. Therefore, graph augmentation is considered of great importance in graph representation learning tasks, as shown in various works [10], [24], [31], [37], [43], [44]. The effective graph augmentation methods can be grouped into two categories. (1) From the perspective of node feature, for example, GDC [31] creates a new graph by utilizing the similarity of adjacent nodes in the graph. AM-GCN [37] constructs a k-nearest neighbor graph based on node features, and JFGCN [43] utilizes the k-nearest neighbor and the k-farthest neighbor to generate an adjacency matrix, in order to keep similar nodes close to each other and different nodes far away from each other.

PA-GCN [24] develops a feature strategy that utilizes the raw graph relationships to avoid introducing noisy node features. (2) From the perspective of graph topology, edge modification [10], [44] builds the second view by randomly perturbing the edge structure of the raw graph. However, these studies [24], [37] consider the fusion of only a single new graph with the raw graph, leading to the underutilization of the complementarity among multiple graphs. In this paper, we reconstruct new graphs using graph augmentation from multiple perspectives and develop a constraint to ensure the complementarity of new graphs with the raw graph.

Multi-Channel Representation Learning of GCN: The multi-channel graph convolutional network model can fully capture the diversity information in the graph structure, thereby improving the performance of the model in various graph structure-based tasks. Multi-channel models typically fuse the representations of different channels to obtain a final embedding representation. For example, there are fusion frameworks based on different multi-order adjacency matrices [34], [35], [36], augmented graph topologies [33], [37], and augmented node features [24]. Notably, the single-channel model CLP-GCN [32] introduces pseudo labeling to achieve excellent performance. Consequently, these studies [24], [33], [37] mostly neglect the importance of pseudo labeling in multi-channel embedding learning. Although MFGCN [22] is a pioneering research to address such a problem, the pseudo labeling only exploits the proximity of the nodes in the embedding space and ignores the similarity of the nodes in the feature space.

III. THE PROPOSED FRAMEWORK: PCM-GCN

A. GCN ENCODER

An undirected graph $\mathcal{G} = (\boldsymbol{A}, \mathcal{V}, \boldsymbol{X})$, $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ is the set of n nodes. The properties of the nodes are described by the feature matrix $\boldsymbol{X} \in \mathbb{R}^{n \times f}$, where n is the total number of nodes, and f is the dimension of their features. $\boldsymbol{A} \in \{0, 1\}^{n \times n}$ is the adjacency matrix, $a_{ij} = 1$ reflects the connectivity between nodes v_i and v_j in the graph with corresponding degree matrix $\boldsymbol{D}$. $\tilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I_n}$ represents the adjacency matrix $\tilde{\boldsymbol{A}}$ for a graph with self-connection and $\tilde{\boldsymbol{D}} = \boldsymbol{D} + \boldsymbol{I_n}$ is the degree matrix of $\tilde{\boldsymbol{A}}$, where $\boldsymbol{I}_n$ is the unit matrix. $\hat{\boldsymbol{A}} = \tilde{\boldsymbol{D}}^{-\frac{1}{2}} \tilde{\boldsymbol{A}} \tilde{\boldsymbol{D}}^{-\frac{1}{2}}$ denotes the symmetrically normalised adjacency matrix with a self-connection. We suppose that C is the number of classes.

The vanilla GCN is proposed by Kipf and Welling [2]. It aggregates information of neighbors by symmetric normalized adjacency matrix $\hat{\boldsymbol{A}}$ to generate new node representation. The simple form of the GCN model can be described by:

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

where $\boldsymbol{W}^{(0)}$ and $\boldsymbol{W}^{(0)}$ denote the trainable weight matrix of the first and second layer of GCN, respectively. σ and $ReLU$ are nonlinear activation functions. $\boldsymbol{\mathcal{Z}}_{GCN}$ is the final prediction for a two-layer GCN.

B. FRAMEWORK OVERVIEW

This section introduces Pseudo-labeling Collaborative Multi-channel Graph Convolutional Network (PCM-GCN). The architecture of PCM-GCN is shown in the following Fig. 1. Our framework encompasses the following key modules.

(1) Graph generation module: graph generation can explore diversity information in graph data. This module develops multiple adjacency matrices by extracting the underlying structure of the nodes from the topology space and feature space. By generating adjacency matrices containing rich underlying information, we enhance support for other modules.

(2) GCN-based multi-channel fusion module: the GCN encoder is specialized for semi-supervised node classification learning. The module is designed to fuse embedding representations in multiple channels into a consistent low-dimensional vector for classification tasks and pseudo labeling module.

(3) Pseudo labeling module: during the training process, this module confidently assigns pseudo-label to unlabeled data. Unlike the traditional multi-channel GCN, we exploit pseudo labeling collaborative embedding representation to obtain a rich node embedding representation.

C. GRAPH GENERATION MODULE

In order to fully exploit the intrinsic information of the graph data distribution, we propose a graph generation module based on the k-nearest neighbor (knn) algorithm and the edge modification (em) algorithm, as in Fig. 2. The two algorithms are designed to obtain more comprehensive graph data in terms of features and topology, as described in detail below.

1) PERTURBATION NODE FEATURES

Firstly, in order to obtain the topology information of the nodes in the feature space, we construct the k-nearest neighbor graphs by means of the node feature matrix $\boldsymbol{X}$. The k-nearest neighbor graphs retain only the connections between nodes and their immediate neighbors, thus reducing the impact of outliers. Specifically, we firstly compute the cosine similarity matrix $\boldsymbol{B} \in \mathbb{R}^{n \times n}$ among n nodes, assuming that x_i and x_j are the feature vectors of the nodes v_i and v_j, and the cosine similarity values between them are formulated as follows:

$$b_{ij} = \frac{< x_i, x_j >}{\parallel x_i \parallel \cdot \parallel x_j \parallel} \tag{2}$$

Then we select the top k similar nodes for each node to update the edges, where k is searched in the range of $[8, 16, 32, \ldots, n]$, and finally we get a set $\boldsymbol{A_f} = \{\boldsymbol{A}_{f_1}, \boldsymbol{A}_{f_2}, \boldsymbol{A}_{f_3}, \ldots, \boldsymbol{A}_{f_n}\}$, and the elements inside $\boldsymbol{A_f}$ denote the corresponding adjacency matrix. Then the feature-based fusion graph $\mathcal{G}_F = (\boldsymbol{\mathcal{A}}_F, \boldsymbol{X})$, where $\boldsymbol{\mathcal{A}}_F$ can be obtained by the following equation:

$$\boldsymbol{\mathcal{A}}_F = \frac{1}{n} \sum\nolimits_{i=1}^{n} \boldsymbol{A}_{f\,i} \tag{3}$$

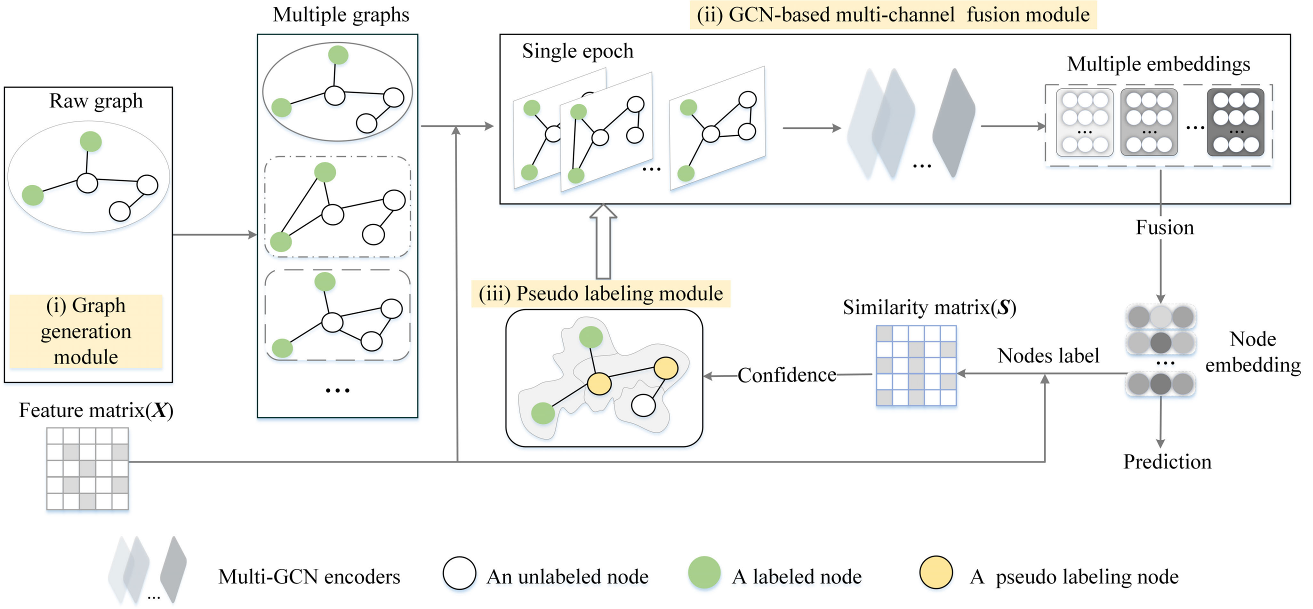


FIGURE 1. PCM-GCN. This framework consists of three main parts: (i) Graph generation module; (ii) GCN-based multi-channel fusion module; (iii) Pseudo labeling module. By introducing pseudo labeling into a multi-channel model, PCM-GCN can effectively enhance the node embedding representation for each channel.

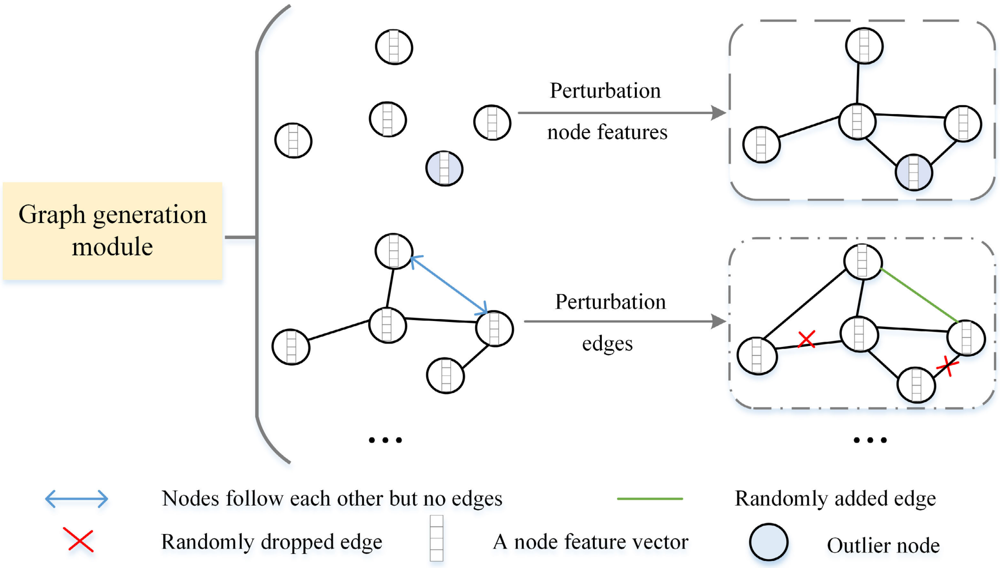


FIGURE 2. The basic concept of the graph generation module.

2) PERTURBATION EDGES

From the input graph adjacency matrix, inspired by the edge modification [10], [44], we not only remove a fraction of the number of edges uniformly and randomly but also add the same number of edges uniformly and randomly at the same time. In this way, we are trying to maximize the preservation of the properties of the raw graph while capturing the information of the multi-order nodes of the graph with the new edges. In our experiments, given an adjacency matrix $\boldsymbol{A}$, we first set a fixed scale P which searches in the range $[0.1, 0.2, 0.3, \ldots, 0.9]$, following the *em* algorithm we can get m new adjacency matrices, denoted as $\boldsymbol{A}_t = \{\boldsymbol{A}_{t_1}, \boldsymbol{A}_{t_2}, \boldsymbol{A}_{t_3}, \ldots, \boldsymbol{A}_{t_m}\}$. Then the edge-based fusion graph $\mathcal{G}_T = (\boldsymbol{\mathcal{A}}_T, \boldsymbol{X})$, where $\boldsymbol{\mathcal{A}}_T$ can be obtained by the following equation:

$$\boldsymbol{\mathcal{A}}_T = \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{A}_{ti} \tag{4}$$

Importantly, equations (4), (5) aim at fusing the multiple graphs into a single augmented graph to enhance the GCN's ability to learn graph data.

D. GCN-BASED MULTI-CHANNEL FUSION MODULE

Given a graph $\mathcal{G} = (\boldsymbol{A}, \mathcal{V}, \boldsymbol{X})$, we use the *knn* and *em* to obtain the graphs $\{\mathcal{G}_F = (\boldsymbol{A}_F, \boldsymbol{X}), \mathcal{G}_T = (\boldsymbol{A}_T, \boldsymbol{X})\}$ in feature space and topology space, respectively. We take vanilla GCN as an encoder to learn graph embedding. We input $\mathcal{G}$ into the two-layer GCN as follows:

$$\mathcal{Z}_1 = (\hat{\boldsymbol{A}} ReLU(\hat{\boldsymbol{A}} \boldsymbol{X} \boldsymbol{W}^{(0)}) \boldsymbol{W}^{(1)}) \tag{5}$$

where $\mathcal{Z}_1$ denotes the embedding result learned from the raw graph.

Similarly, $\mathcal{G}_F$ and $\mathcal{G}_T$ are input into the independent GCN to obtain graph embedding $\mathcal{Z}_2$ and $\mathcal{Z}_3$, respectively. For the three graph embeddings, namely $\mathcal{Z}_1$, $\mathcal{Z}_2$ and $\mathcal{Z}_3$, we use the fusion mechanism to determine the final embedding output $\mathcal{Z}$.

$$\mathcal{Z} = \sigma(\frac{1}{3} \textstyle\sum_{i=1}^{3} \mathcal{Z}_i) \tag{6}$$

We suppose that the subset $\mathcal{V}_L = \{v_1, v_2, \ldots, v_L\}$ of node set $\mathcal{V}$ is labeled training set, and the corresponding labels are $\mathcal{Y}_L = \{y_1, y_2, \ldots, y_L\}$. In addition, $\mathcal{V}_U$ is is the unlabeled set, we can obtain $\mathcal{V} = \mathcal{V}_L \cup \mathcal{V}_U$. For each node $v_l \in \mathcal{V}_L$ has two types of labels: real label y_l and predicted label $\mathcal{Z}_l$. Then the cross entropy error for node classification over all labeled nodes is represented as:

$$\mathcal{L}_0 = -\textstyle\sum_{v_l \in \mathcal{V}_L} \sum_{t=1}^{C} y_l \ln \mathcal{Z}_{lt} \tag{7}$$

For $\mathcal{Z}_2$ and $\mathcal{Z}_3$, they are the embedding representations of nodes, respectively. To ensure that $\mathcal{Z}_2$ and $\mathcal{Z}_3$ are closer to $\mathcal{Z}_1$ in the embedding space. Here, we design a consistency constraint to ensure their commonality. Consistency indicates that the augmented graph is similar and close to the raw graph in the embedding space, which yields the following constraint:

$$\begin{aligned} \mathcal{L}_1 = & \left\| \mathcal{Z}_{2nor} \cdot \mathcal{Z}_{2nor}^T - \mathcal{Z}_{1nor} \cdot \mathcal{Z}_{1nor}^T \right\|_F^2 + \\ & \left\| \mathcal{Z}_{3nor} \cdot \mathcal{Z}_{3nor}^T - \mathcal{Z}_{1nor} \cdot \mathcal{Z}_{1nor}^T \right\|_F^2 \end{aligned} \tag{8}$$

where $\mathcal{Z}_{1nor}, \mathcal{Z}_{2nor}$ and $\mathcal{Z}_{3nor}$ denote the embedding matrices $\mathcal{Z}_1, \mathcal{Z}_2$ and $\mathcal{Z}_3$ are normalised by the L_2-normalization.

E. PSEUDO LABELING MODULE

Pseudo labeling plays an important role in alleviating the lack of labels in semi-supervised learning classification tasks. The features of nodes with the same label tend to be similar, so we use the feature cosine similarity between two nodes to mark unlabeled nodes with pseudo labels. In this paper, we take two perspectives: (1) the similarity of nodes in the feature space; (2) the proximity of nodes in the embedding space.

Initially, we estimate the pseudo label with the cosine similarity between $\mathcal{V}_L$ and $\mathcal{V}_U$. The element s_{ij} in similarity matrix $\boldsymbol{S} \in \mathbb{R}^{n \times n}$ can be calculated by the sum of the embedding space matrix $\mathcal{Z}$ and the node feature space matrix $\boldsymbol{X}$, which is expressed by the following formula:

$$s_{ij} = \frac{\langle \mathcal{Z}_i, \mathcal{Z}_j \rangle}{\|\mathcal{Z}_i\| \cdot \|\mathcal{Z}_j\|} + \frac{\langle \boldsymbol{X}_i, \boldsymbol{X}_j \rangle}{\|\boldsymbol{X}_i\| \cdot \|\boldsymbol{X}_j\|} \tag{9}$$

where $\langle , \rangle$ denotes the inner product between vectors, and $\|\cdot\|$ is the operation of vector lengths, s_{ij} stands for the cosine similarity between v_i and v_j. For an unlabeled node $v_i \in \mathcal{V}_U$ and every labeled nodes $v_j \in \mathcal{V}_L$, compute all s_{ij} between them, i.e., the maximum value of s_{ij} indicates that the two nodes most likely belong to the same class. The maximum value can be achieved by comparing $\{s_{ij}\}_{j=1}^{L} = \{s_{i1}, s_{i2}, \ldots, s_{iL}\}$, then the pseudo labeling loss function can be expressed as following:

$$\hat{y}_i = y_j \quad \text{if } \max \{s_{ij}\}_{j=1}^{L} \geq \delta \tag{10}$$

where δ is an adjustable threshold, $\hat{y}_i \in \hat{Y}_i$ is the pseudo-label of v_i and $y_j \in \mathcal{Y}_j$ is the true label of v_j, the labels of Y_j are distributed to $\hat{Y}_i$ as pseudo-label.

Meanwhile, the index a_i can be found by the following formula:

$$a_i = \begin{cases} 1, & \text{if } \max \{s_{ij}\}_{j=1}^{L} \geq \delta, \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

Then the loss function for unlabeled samples is formulated as:

$$\mathcal{L}_2 = -\textstyle\sum_{v_i \in \mathcal{V}_U} \sum_{t=1}^{C} a_i \cdot \hat{y}_i \ln \mathcal{Z}_{it} \tag{12}$$

Explain that for each $v_i \in V_U$, the pseudo labeling is $\hat{Y}_i$ and the predicted label is $\mathcal{Z}_{it}$. Therefore, the final loss function of the model is expressed as follows:

$$\mathcal{L} = \mathcal{L}_0 + \mathcal{L}_1 + \lambda \mathcal{L}_2 \tag{13}$$

where λ is a balance hyper-parameter.

Algorithm 1 shows the entire training process of the model.

Algorithm 1 The Training Framework of PCM-GCN

```
Input: Feature matrix $\boldsymbol{X}$; adjacency matrices $\boldsymbol{A}$; epochs $\mathcal{N}$;
    hyper-parameters $n, m, P, \delta$ and $\lambda$;
Output: The sample class prediction matrix $\mathcal{Z}$;
Initialize hyper-parameters: $n, m, P, \lambda$, and $\delta$;
Perturbing the node feature of $\boldsymbol{X}$ creates a set $\boldsymbol{A}_f$ via $k$-nearest neighbor;
Perturbing the edge structure of $\boldsymbol{A}$ creates a set $\boldsymbol{A}_t$ via edge modification;
for int epochs = 0 to $\mathcal{N} - 1$ do
    Calculate $\mathcal{A}_F$ and $\mathcal{A}_T$ by Eq.(3), Eq.(4);
    Calculate $\mathcal{Z}_1, \mathcal{Z}_2$ and $\mathcal{Z}_3$ by Eq.(5);
    Calculate $\mathcal{Z}$ by Eq.(6);
    Calculate $\mathcal{L}_0$, $\mathcal{L}_1$, $\mathcal{L}_2$ respectively by Eq.(7), Eq.(8), and Eq.(12);
    Calculate total loss: $\mathcal{L} = \mathcal{L}_0 + \mathcal{L}_1 + \lambda \mathcal{L}_2$;
end for
return $\mathcal{Z}$
```

IV. EXPERIMENTS

In this section, we test our proposed model against several popular graph-based semi-supervised node classification models. Then, we perform ancillary experiments to evaluate the effects of hyper-parameter and the performances of the components of PCM-GCN. The visualisation of model accuracy further demonstrates the effectiveness of PCM-GCN.

A. DATASETS

In this study, we utilize five real benchmark datasets. The citation networks Cora, Citeseer, and Pubmed, where the nodes represent papers and the edges represent citation relationships among papers. ACM is a dataset of paper relationships, where the nodes represent papers and two papers are considered related if they share at least one author. Coauthor-cs is a dataset of coauthorship relationships, where each node represents an author, and there exists an edge between two authors if they have collaborated on at least one paper. The detailed statistics of these five benchmark datasets are shown in Table. 1.

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

No.	Dataset	Node	Edges	Features	Classes	Training
1	Cora	2708	5429	1433	7	35/70/140
2	Citeseer	3327	4732	3703	6	30/60/120
3	Pubmed	19717	443328	500	3	15/30/60
4	ACM	3025	13128	1870	3	15/30/60
5	Coauthor-cs	18333	81894	6805	15	75/150/300

B. BASELINES

We compare our proposed PCM-GCN with state-of-the-art semi-supervised classification learning methods, including the following five categories:

- Base encoder: GCN [2], SGC [4], GAT [1].
- Sampling-based encoder: FastGCN [3].
- Multi-scale information fusion-based encoder: N-GCN [23], IGCL [11].
- Graph generation fusion-based encoder: MOGCN [34], PA-GCN [24].
- Pseudo labeling-based encoder: MFGCN [22], LaenNet [21].

We run the proposed model 20 times and record the average accuracy. For the other methods, we operate the code provided by the authors and record the average results.

C. EXPERIMENT SETTINGS

In our study, 500 samples are taken as validation set data and 1000 samples are utilized as test set samples. For these five datasets, we use only the labeled training set to train the PCM-GCN model. The Adam optimizer with a learning rate of 0.01 is employed to train the proposed PCM-GCN. In these five datasets, the value of weight decay is set to 5e-4, the number of hidden neurons is selected from [8, 16, 32], and the dropout parameter is placed at 0.8 to prevent over-fitting. In addition, δ is set at the range of $\{0.9, 1\}$ in order to maximize the performance of pseudo labeling module. In our experiment, m and n take the same value and the balance parameters m and n take values in the range $\{1, 2, 3, 4, 5\}$, we perform a sensitivity study for these two parameters.

TABLE 2. Summary of classification accuracy (%). The best results are highlighted.

Datasets	Method	Training		
Cora		35	70	140
	GAT	76.30	71.10	82.30
	GAT+LaenNet	75.70	81.00	83.30
	GAT+OURS	**79.20**	**81.90**	**84.50**
	SGC	69.13	74.41	77.64
	SGC+LaenNet	73.37	78.26	81.49
	SGC+OURS	**77.90**	**81.60**	**81.90**
Citeseer		30	60	120
	GAT	65.40	66.80	72.50
	GAT+LaenNet	70.50	67.70	72.30
	GAT+OURS	**71.00**	**72.10**	**74.30**
	SGC	60.11	67.18	70.29
	SGC+LaenNet	63.95	**69.00**	71.44
	SGC+OURS	**66.70**	65.30	**72.40**
Pubmed		15	30	60
	SGC	69.76	73.10	**78.90**
	SGC+LaenNet	71.32	74.64	78.24
	SGC+OURS	**72.50**	**76.20**	78.20
ACM		15	30	60
	GAT	87.20	87.80	90.20
	GAT+OURS	**92.00**	**93.20**	**93.70**
	SGC	44.50	70.41	86.32
	SGC+OURS	**89.10**	**91.60**	**91.80**
Coauthor-cs		15	30	60
	SGC	75.32	75.32	72.72
	SGC+LaenNet	90.20	90.90	92.00
	SGC+OURS	**90.40**	**92.90**	**92.80**

D. EXPERIMENT RESULTS

1) NODE CLASSIFICATION

The average accuracy (%) of the node classification of ours and baseline are summarized in Table 3, where the bold value represents the best result. We observe the following results:

(1) Compared to the baseline described above, PCM-GCN is confirmed to achieve the best results on all five datasets. Notably, PCM-GCN shows a significant advantage on the Citeseer, ACM, and Coauthor-cs datasets.

TABLE 3. Accuracy(%) of node classification tasks. The best results are highlighted.

	Training	GCN	GAT	SGC	LaenNet	FastGCN	IGCL	N-GCN	MOGCN	MFGCN	PA-GCN	**PCM-GCN**
Cora	35	77.49	76.30	69.30	75.07	71.10	76.60	63.40	72.50	58.60	72.10	**79.91**
	70	76.68	71.10	74.41	79.33	71.58	77.96	78.80	78.90	68.50	73.50	**82.30**
	140	80.62	82.30	77.64	81.54	80.7	80.70	83.00	83.10	75.10	83.60	**83.69**
Citeseer	30	60.11	65.40	63.50	67.16	63.00	64.24	57.40	62.40	68.30	64.30	**69.30**
	60	67.19	66.80	67.95	71.15	67.80	69.96	68.20	68.10	70.02	67.20	**73.01**
	120	70.29	72.50	70.01	72.59	70.70	72.10	72.20	72.40	71.90	70.40	**74.40**
Pubmed	15	69.76	69.94	67.27	71.80	67.64	65.20	62.40	66.17	70.60	63.50	**72.80**
	30	73.10	73.54	74.69	75.44	73.88	68.10	58.10	73.95	73.80	73.50	**75.70**
	60	78.90	79.00	77.48	78.92	79.40	**80.8**	79.50	79.20	79.10	79.30	80.10
ACM	15	65.60	87.20	44.50	-	86.80	72.20	64.30	68.10	87.90	69.70	**92.90**
	30	79.60	87.80	70.41	-	87.70	87.30	76.50	87.50	90.90	87.20	**93.00**
	60	78.36	90.20	86.32	-	89.20	89.20	88.00	90.10	92.20	90.90	**93.70**
Coauthor-cs	75	87.70	84.29	88.10	90.00	89.3	85.6	-	87.00	89.00	82.40	**91.90**
	150	89.60	87.22	90.89	90.6	89.9	86.5	-	90.50	91.20	84.40	**92.70**
	300	90.34	88.66	91.55	91.4	91.1	88.7	-	91.20	92.30	87.00	**93.40**

TABLE 4. Accuracy(%) of ablation experiments on five datasets.

Datasets	Cora			Citeseer			Pubmed			ACM			Coauthor-cs		
	35	70	140	30	60	120	15	30	60	15	30	60	75	150	300
GCN	77.49	76.68	80.62	60.11	67.19	70.29	69.76	73.10	78.90	65.60	79.60	78.36	87.70	89.60	90.34
GCN+$\mathcal{A}$	77.00	78.90	82.40	63.00	67.80	71.70	75.10	76.65	79.40	85.10	84.00	90.30	89.76	91.30	92.11
GCN+$\mathcal{B}$	77.13	78.40	82.50	61.10	68.20	71.50	75.42	76.40	79.00	89.90	91.56	91.90	89.61	91.96	91.81
PCM-GCN	79.90	82.30	83.70	69.30	73.00	74.70	72.81	75.70	80.10	91.90	92.70	93.40	91.90	92.70	93.40

(2) PCM-GCN outperforms multi-scale information fusion based encoder (MOGCN, N-GCN, PA-GCN, IGCL, GCN+LaenNet), multi-view fusion based encoder(MFGCN) and sampling based encoder(FastGCN) on most of the datasets. The explanation is that our proposed model reconstructs the graph in terms of raw graph topology and node features, allowing the model to capture more reliable information from different graphs. At the same time, the fusion of multi-channel information can significantly improve the representation of the model for downstream tasks.

(3) When labeled samples are sparse, such as when there are only 5 or 10 samples per class, PCM-GCN has been proven to be effective, i.e., PCM-GCN maintains a fairly good classification performance. Especially for the ACM dataset, our proposed model relieves the problem of rapid degradation of classification performance due to label sparsity. This is attributed to our pseudo labeling module that generates feasible pseudo-labels for a large number of unlabeled samples in semi-supervised learning, thereby meeting the label data requirements for training comprehensive node embedding representation.

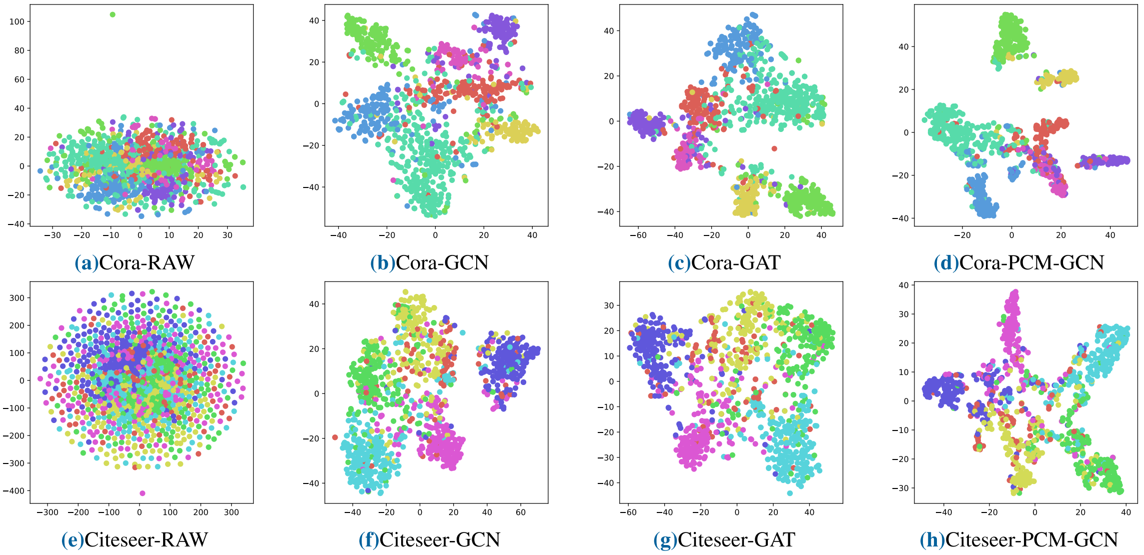
2) FRAMEWORK STUDY

As previously recounted, PCM-GCN can easily incorporate different GCNs frameworks without imposing restrictions. In this section, in addition to GCNs, we incorporate basic encoders (SGC, GAT) into the PCM-GCN framework, denoted as GAT+OURs and SGC+OURs, respectively. Defining GAT, SGC, GAT+LaenNet, and SGC+LaenNet as the baselines, we evaluate their classification accuracy under sparse labels, and the results are presented in Table 2.

From Table 2, PCM-GCN can enhance GCN on most datasets. In particular, for Cora, ACM, and Coauthor-cs, our proposed model shows significant improvement at different labeled rates. For example, PCM-GCN can improve GAT by up to 10.8 % on Cora when the number of labeled nodes per class is 10. The above observation shows the scalability of PCM-GCN on different encoders.

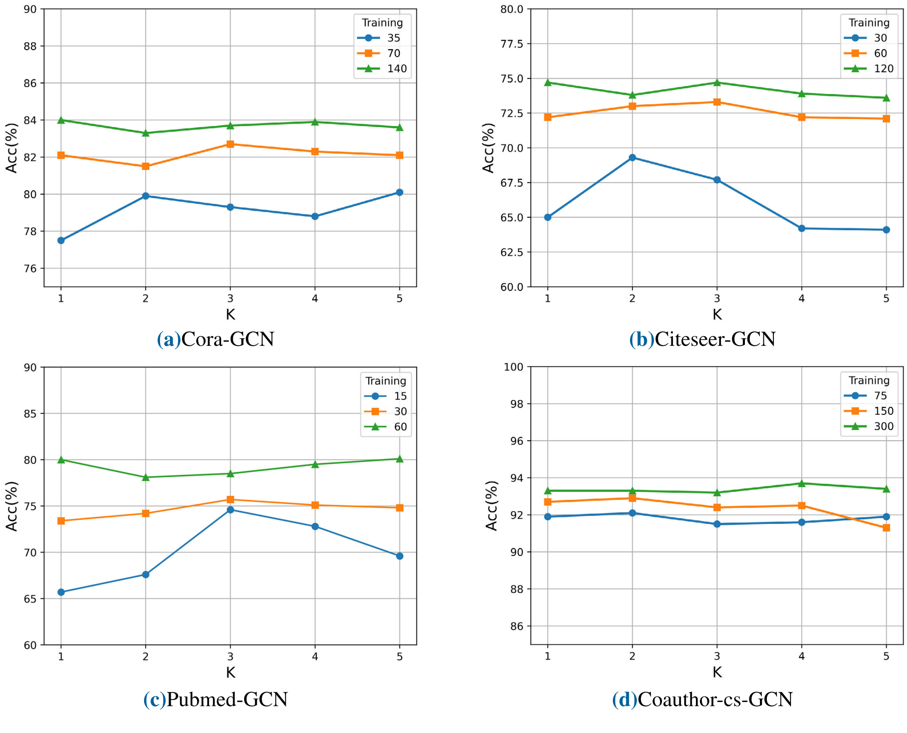
3) ABLATION EXPERIMENT

In order to clearly explain the validity of the graph generation module and the pseudo labeling module, we conducted ablation experiment to show the contribution of each component.


(a)Cora-RAW (b)Cora-GCN (c)Cora-GAT (d)Cora-PCM-GCN

(e)Citeseer-RAW (f)Citeseer-GCN (g)Citeseer-GAT (h)Citeseer-PCM-GCN

FIGURE 3. **The visualization of classification results of GCN, GAT, and PCM-GCN on Cora, Citeseer datasets with 140, 120 labeled samples.**



(a)Cora-GCN (b)Citeseer-GCN

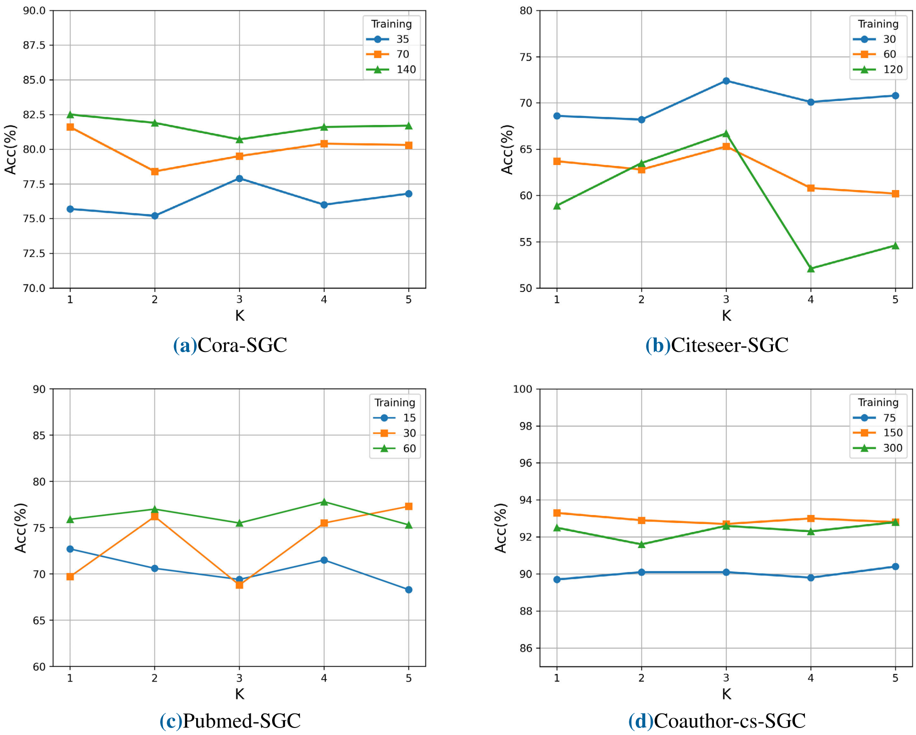
(c)Pubmed-GCN (d)Coauthor-cs-GCN

FIGURE 4. **The performance of GCN+OURs with varied hyper-parameter *K*.**

Two ablation experiments are designed as follows and the experimental results are shown in Table 4.

- **GCN #$\mathcal{A}$**: GCN #$\mathcal{A}$ denotes a variant of GCN which is the base GCN fused with the graph generation module; the hyper-parameters n and m take equal values in the experiments, and we search for the highest accuracy in the range $\{1, 2, 3, 4, 5\}$ as the experimental result.
- **GCN #$\mathcal{B}$**: GCN #$\mathcal{B}$ represents a method that is based on the pre-training of the vanilla GCN combined with the pseudo labeling module to form a learner.

For **GCN #$\mathcal{A}$**: It improves the classification accuracy at different label rates compared to the original GCN. For example, observing the ACM and Coauthor-cs datasets, the GCN with graph generation mechanism improves the



(a)Cora-SGC

(b)Citeseer-SGC

(c)Pubmed-SGC

(d)Coauthor-cs-SGC

FIGURE 5. **The performance of SGC+OURs with varied hyper-parameter K.**

classification accuracy by 19.5%, 14.4%, 11.94% and 2.06%, 1.7%, 1.87% respectively.

For **GCN** #$\mathcal{B}$: Compared to GCN, GCN with pseudo labeling module performs better at low label rates. Specifically, when the training set of ACM is 30, the improvement is 24.3%. For Pubmed with a sample size of 15, the pseudo labeling module helps GCN to improve by 4.66%. This confirms that our pseudo labeling eases the label sparsity problem in the semi-supervised tasks.

It is observed that PCM-GCN outperforms GCN #$\mathcal{A}$ and GCN #$\mathcal{B}$ on the five datasets, which confirms that the combination of the graph generation module and the pseudo labeling module is effective.

4) CLASSIFICATION RESULT VISUALIZATION

In order to demonstrate the effectiveness of our proposed model more intuitively, we perform a visualisation task (distribution of raw data and classification results of GCN, GAT and PCM-GCN) on the Cora and Citeseer datasets using the t-SNE [45] algorithm. As can be seen in Fig. 3, the distribution of the raw data is irregular. Meanwhile, the visualisation results of GCN and GAT are not satisfying because there is no clear boundary between different classes. Obviously, the proposed model performs better due to its larger inter-class distance and smaller intra-class distance.

5) PARAMETER SENSITIVITY STUDY

In this section, we investigate the sensitivity of the balance parameters n and m to the experimental results over four datasets. the values of n and m are set to $\{1, 2, 3, 4, 5\}$, and for simplicity we use K in place of n and m. Fig. 4 and Fig. 5 illustrate the effect of different values of K on the node classification results.

It is observed that classification accuracy at different label rates tends to obtain best at $K > 1$, which confirms the validity of our proposal to integrate multi-graphs into one enhanced graphs.

V. CONCLUSION

In this paper, we propose an end-to-end GCNs framework for semi-supervised node classification tasks. This framework develops three modules: (1) graph generation module; (2) multi-channel fusion module; (3) pseudo labeling module. The graph generation module utilizes the topology and features of the raw graph to generate multiple graphs individually, and the multiple graphs are input to separate GCN encoder. Then, the multi-channel fusion module aims to fuse the embedding representations of multiple GCNs, achieving joint learning of important information hidden in the topology and features. Finally, the pseudo labeling module utilizes unlabeled node information to enhance the embedding representation, achieving more effective node embedding. Extensive experiments show that the proposed model can obtain good performance even with a small number of labels. Furthermore, we have showed the flexibility and effectiveness of PCM-GCN for introducing different GCNs models in our framework study. Experiments on five publicly available datasets have demonstrated that our proposed model outperforms other state-of-the-art models in semi-supervised node classification. For future work, we aim to explore the modeląŕs potential for other graph-related tasks.

REFERENCES

[1] P. Velickovic, G. Cucurull, A. Casanova, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. 6th Int. Conf. Learn. Represent.*, 2018.

[2] T.-N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. 5th Int. Conf. Learn. Represent.*, 2017.

[3] J. Chen, T.-f. Ma, and C. Xiao, "FastGCN: Fast learning with graph convolutional networks via importance sampling," in *Proc. 6th Int. Conf. Learn. Represent.*, 2018.

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

[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, no. 7, 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, no. 6, 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, doi: 10.1007/s13042-023-02013-2.

[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, no. 4, 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] D. H. Nguyen and K. Tsuda, "On a linear fused Gromov–Wasserstein distance for graph structured data," *Pattern Recognit.*, vol. 138, Jun. 2023, Art. no. 109351.

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

[20] M. Chatzianastasis, J. Lutzeyer, G. Dasoulas, and M. Vazirgiannis, "Graph ordering attention networks," in *Proc. AAAI Conf. Artif. Intell.*, Jun. 2023, vol. 37, no. 6, pp. 7006–7014.

[21] 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.

[22] Y. Yang, Y. Sun, F. Ju, S. Wang, J. Gao, and B. Yin, "Multi-graph fusion graph convolutional networks with pseudo-label supervision," *Neural Netw.*, vol. 158, pp. 305–317, Jan. 2023.

[23] 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 Uncertainty Artif. Intell. Conf.*, 2020, pp. 841–851.

[24] 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.

[25] E.Chien, J. Peng, P. Li, and O. Milenkovic, "Adaptive universal generalized pagerank graph neural network," in *Proc. 9th Int. Conf. Learn. Represent.*, 2021.

[26] H. Dong, L. Li, D. Tian, Y. Sun, and Y. Zhao, "Dynamic link prediction by learning the representation of node-pair via graph neural networks," *Expert Syst. Appl.*, vol. 241, May 2024, Art. no. 122685.

[27] 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.

[28] 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.

[29] M. Zou, Z. Gan, R. Cao, C. Guan, and S. Leng, "Similarity-navigated graph neural networks for node classification," *Inf. Sci.*, vol. 633, pp. 41–69, Jul. 2023.

[30] Y. Yi, X. Lu, S. Gao, A. Robles-Kelly, and Y. Zhang, "Graph classification via discriminative edge feature learning," *Pattern Recognit.*, vol. 143, Nov. 2023, Art. no. 109799.

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

[32] 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.

[33] 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.

[34] 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.

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

[36] 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.

[37] X. Wang, M. Zhu, D. Bo, P. Cui, C. Shi, and J. Pei, "AM-GCN: Adaptive multi-channel graph convolutional networks," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 1243–1253.

[38] D.-H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Proc. Int. Conf. Mach. Learn.*, vol. 3, 2013, pp. 896–902.

[39] M.-N Rizve, K. Duarte,Y.-S Rawat, and M. Shah, "In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi- supervised learning," in *Proc. 9th Int. Conf. Learn. Represent.*, 2021.

[40] L. Zhong, J. Lu, Z. Chen, N. Song, and S. Wang, "Adaptive multi-channel contrastive graph convolutional network with graph and feature fusion," *Inf. Sci.*, vol. 658, Feb. 2024, Art. no. 120012.

[41] 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.

[42] H. Duan, C. Xie, B. Li, and P. Tang, "Self-supervised contrastive graph representation with node and graph augmentation," *Neural Netw.*, vol. 167, pp. 223–232, Oct. 2023.

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

[44] M. Jin, Y. Zheng, Y.-F. Li, C. Gong, C. Zhou, and S. Pan, ''Multi-scale contrastive Siamese networks for self-supervised graph representation learning,'' in *Proc. 13th Int. Joint Conf. Artif. Intell.*, Aug. 2021, pp. 1477–1483.

[45] L. van der Maaten and G. Hinton, ''Visualizing data using t-SNE,'' *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

MENGNAN PANG received the bachelor's degree from the School of Computer Science and Technology, Luoyang University of Technology. He is currently pursuing the master's degree with the School of Computer, Jiangsu University of Science and Technology. His current research interests include machine learning and rough set.

XUN WANG received the B.S. degree from Jiangsu University of Science and Technology and the M.E. degree from Jiangsu University. She is currently an Associate Professor with the School of Computer, Jiangsu University of Science and Technology. Her research interests include data mining, big data processing, and machine learning.

JINGJING SONG received the M.E. degree from Jiangsu University of Science and Technology and the Ph.D. degree from Macau University of Science and Technology. She is currently an Associate Professor with the School of Computer, Jiangsu University of Science and Technology. Her research interests include machine learning and rough set.

PINGXIN WANG received the B.S. degree from Xinyang Normal University, the M.E. degree from Nanjing University of Science and Technology, and the Ph.D. degree from Nanjing University of Aeronautics and Astronautics. He is currently an Associate Professor with the School of Computer, Jiangsu University of Science and Technology. His current research interests include numerical algebra and rough set.