2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

Multi-Channel Fusion Graph Convolutional Networks with pseudo-label for Semi-Supervised Node Classification

Guang Yang[0009-0003-4361-2568] , Shiwen Sun ✉ [0009-0001-3236-0508] , Zhouhua Shi[0009-0009-9331-9179]

Tianjin University of Technology, School of Computer Science and Engineering
300380, Tianjin,china
sunsw@tjut.edu.cn

Abstract. Graph Convolutional Networks (GCNs) have shown great promise in semi-supervised node classification tasks. However, existing Graph Convolutional Networks (GCNs) face two key challenges: 1) While addressing the limitations of incomplete or noisy graph structures, the structural information of the graph remains underutilized; 2) the scarcity of labeled data, limiting the ability to learn comprehensive embeddings. In response to these challenges, we propose a novel Multi-channel Fusion Graph Convolutional Networks with pseudo-label, which learn a connected embedding by fusing the multi-channel graphs information and node features. First, to explore the latent information within the original data, we design a graph generation module to extend and reconstruct the original data into multiple graphs. Meanwhile, a multi-channel approach is employed to embed and fuse these graphs, capturing the complementarity across different channels. Second, to address the issue of label sparsity, we design a confidence propagation-based information gain filtering module to generate high-quality pseudo-labels.
Extensive experiments on three benchmark datasets demonstrate that our method outperforms other approaches.

Keywords: Graph convolutional networks, Multi-channel, Pseudo labeling, Semi-supervised, Classification learning.

1 Introduction

Graph Neural Networks (GNNs) and their variants have demonstrated outstanding performance across various graph-based tasks, including node classification [1], [2], [3], [4], [5], [6], [7], [8], link prediction [9], [10] and graph classification [11]. GNNs employ a computational paradigm known as message passing [12], in which feature aggregation serves as a key component. In this process, at every convolution layer, nodes aggregate features from their connected neighbors, propagating information through the network topology. This iterative process enriches node embeddings, thereby enhancing the model's effectiveness in processing graph-form data.

Node classification is a core task in GNNs, widely used in areas like social network analysis, bioinformatics, and recommendation systems. Although GNNs and their variations have shown notable success in semi-supervised node classification, most existing methods heavily rely on predefined graph structures (i.e., adjacency matrices) to learn node embeddings. However, the edges (relationships) in these graphs are typically manually defined, which may not fully capture the latent patterns and global information in the data. This restriction limits the expressive potential of GNNs. In response, recent studies [13], [14], [15],[16] have introduced dual-channel or multi-channel frameworks to enhance the modeling capability of GNNs for node relationships. For instance, AM-GCN [17] generates a k-nearest neighbor graph from the node feature matrix and simultaneously extracts node embeddings from the kNN graph, the original graph, and their combination, thereby enhancing the model's capacity to capture both local and global structural information. Additionally, other studies [14] have proposed constructing multiple adjacency matrices to enhance the connectivity between nodes, facilitating more efficient message passing in GCNs.

Although these methods contribute to the success of GNNs in semi-supervised node classification, several challenges remain. First, the alternative graph structures constructed in existing models are often simplistic, limiting their ability to capture complex structural information and reducing the model's expressive power. Second, limited labels hinder effective GNN training. To overcome this limitation, some studies have introduced pseudo-label generation mechanisms, such as MFGCN [18], which employs a similarity-based pseudo-label generation method. However, such approaches may introduce noisy pseudo-labels, potentially degrading the overall model performance.

This paper proposes a multi-channel fusion graph convolutional network with pseudo-labels for semi-supervised node classification tasks. In this model, the graph generation module processes the original graph data using two graph construction algorithms: multi-scale feature fusion and edge-augmented graph fusion, resulting in multiple complementary feature graphs. Next, we design a multi-channel fusion module that applies GCN encoders for convolution operations and fusion of these graphs. To ensure consistency in the embeddings across different graphs during training, we introduce a consistency function. Finally, we propose a confidence propagation-based information gain filtering mechanism for selecting and generating pseudo-labels, This mechanism facilitates node classification. Our primary contributions are outlined below:

1. Considering that graphs generated from a single perspective may not effectively support downstream tasks, we design a method to generate multiple graphs for processing by the GCN encoder, enabling the acquisition of richer node representations.
2. We have designed an end-to-end Graph Neural Network (MCF-GCN) model employing multi-channel encoding to process multiple generated graphs while enforcing consistency constraints, ultimately fusing them to generate node representations.
3. To overcome the scarcity of labeled data, we propose generating pseudo-labels with confidence propagation and information gain filtering, enabling the selection of more reliable pseudo-labels.

2 Preliminaries

2.1 Graph Convolutional Networks

A common type of GNN is the Graph Convolutional Network (GCN), developed by Kipf and Welling [3]. In a GCN, node representations are updated by aggregating information from their neighbors using a symmetrically normalized adjacency matrix, denoted as $\widehat{\mathbf{A}}$. The symmetrically normalized adjacency matrix with self-loops is defined as:

$$\widehat{\mathbf{A}} = \widetilde{\mathbf{D}}^{-\frac{1}{2}}\widetilde{\mathbf{A}}\widetilde{\mathbf{D}}^{-\frac{1}{2}} \tag{1}$$

Here, $\widetilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ represents the adjacency matrix with self-loops added, and $\widetilde{\mathbf{D}}$ is the degree matrix of $\widetilde{\mathbf{A}}$. The GCN aggregates information using this normalized matrix, and its two-layer formulation is expressed as:

$$\mathbf{Z}_{\mathrm{GCN}} = \sigma\left(\widehat{\mathbf{A}}\,\mathrm{ReLU}\left(\widehat{\mathbf{A}}\mathbf{X}\mathbf{W}^{(0)}\right)\mathbf{W}^{(1)}\right) \tag{2}$$

where $\mathbf{W}^{(0)}$ and $\mathbf{W}^{(1)}$ represent the learnable weight matrices for the first and second layers of the GCN, respectively. σ and ReLu are nonlinear activation functions. $\mathbf{Z}_{\mathrm{GCN}}$ represents the ultimate result from the two-layer GCN.

3 METHOD

In this section, we present the Multi-Channel Fusion Graph Convolutional Networks with pseudo-label. The architecture of MCF-GCN is depicted in **Fig. 1**. Our approach is composed of the following components:

(1) Graph Generation Module: The module can uncover diverse information within graph data by leveraging both graph topology and node feature information to extract rich latent information, thus enabling graph reconstruction and providing edge information for supporting subsequent modules.

(2) GCN-based Mutil-channel Fusion Module: This module fuses node embeddings from muti-channels using GCNs. The fused embeddings are then used for node classification.

(3) Pseudo-Label Generation Module: To address the problem of label sparsity, We propose a novel mechanism for selecting pseudo-labels. We first perform an

initial screening through confidence propagation and then apply an information filtering mechanism to identify pseudo-labels with higher reliability.

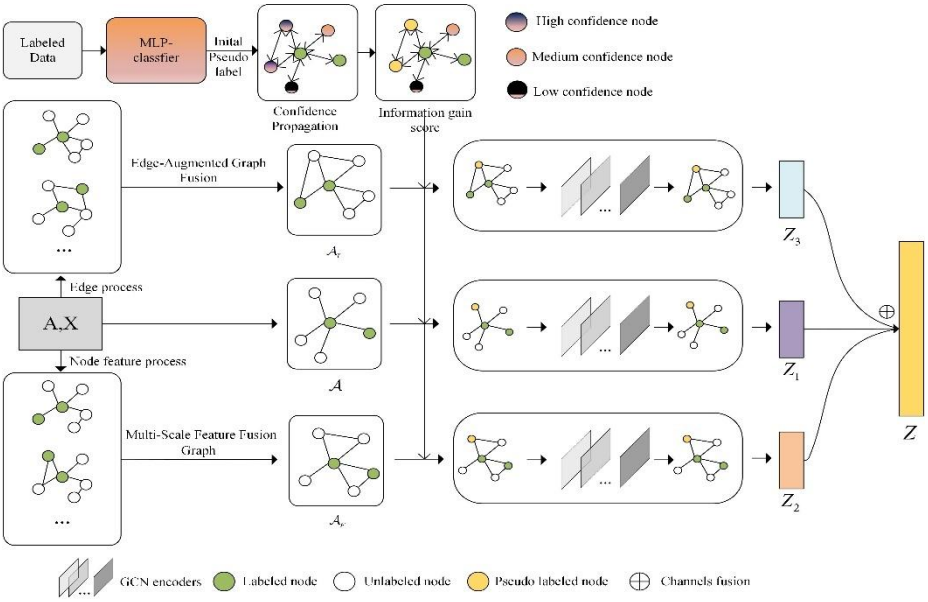


Fig. 1: The framework of MCF-GCN: The model consists of three key modules. The graph generation module constructs an edge-augmented fusion graph $\mathcal{A}_T$ and a multi-scale feature fusion graph $\mathcal{A}_F$ from the original data. A multi-channel independent GCN extracts and integrates multi-channel information from the original graph $\mathcal{A}$, the generated graphs $\mathcal{A}_T$ and $\mathcal{A}_F$. The pseudo-label generation module supe-vises training by assigning high-quality pseudo-labels to unlabeled nodes.

3.1 Graph Generation

Edge-Augmented Graph Fusion

From the input graph adjacency matrix, inspired by the edge modification, We randomly remove a fraction of the edges and simultaneously add an equal number of edges in a uniform manner. 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 $\mathcal{A}$, we first set a fixed scale P which searches in the range [0.1,0.2,0.3, … ,0.9]. Following the EM algorithm, we can get m new adjacency matrices, denoted as $\mathbf{A}_t = \{\mathbf{A}_{t1}, \mathbf{A}_{t2}, \mathbf{A}_{t3}, \dots, \mathbf{A}_{tn}\}$. Then, the edge-based fusion graph $\mathcal{G}_T = (\mathcal{A}_T, X)$, where $\mathcal{A}_T$ can be obtained by the following equation:

$$\mathcal{A}_T = \frac{1}{n}\sum_{i=1}^{n} \mathbf{A}_{t_i} \tag{3}$$

Multi-Scale Feature Fusion Graph

To grasp the topological relationships between nodes in feature space, we generate a graph based on cosine similarity from the node feature matrix **X**. This approach helps identify connections between nodes with similar characteristics while minimizing the impact of outliers.

We begin by computing the cosine similarity matrix$S \in \mathbb{R}^{n\times n}$ for n nodes. Let $\mathbf{x}_i$ and $\mathbf{x}_j$ represent the feature vectors of nodes v_i and v_j, respectively. The cosine similarity $S_{i,j}$ between these two nodes is calculated as follows:

$$S_{i,j} = \frac{\langle \mathbf{x}_i, \mathbf{x}_j \rangle}{\| \mathbf{x}_i \| \cdot \| \mathbf{x}_j \|} \tag{4}$$

Then, we select the top k similar nodes for each node to update the edges, where k is randomly chosen from the range of $[8,16,32,\dots,S]$. Finally, we obtain a set $\mathbf{A}_f = \{\mathbf{A}_{f1}, \mathbf{A}_{f2}, \mathbf{A}_{f3} \dots, \mathbf{A}_{fm}\}$, where the elements inside $\mathbf{A}_f$ represent the corresponding adjacency matrices.

Then, to obtain multi-scale neighborhood information and reduce the impact of noise, the feature-based fusion graph is defined as $G_F = (\mathbf{A}_F, \mathbf{X})$, where $\mathbf{A}_F$ is obtained by the following equation:

$$\mathcal{A}_F = \frac{1}{m}\sum_{i=1}^{m} \mathbf{A}_{fi} \tag{5}$$

3.2 GCN-based Mutil-channel Fusion Module

Given a graph $\mathcal{G} = (\mathcal{A}, \mathcal{V}, \mathcal{X})$, we use the edge-augmented fusion and multi-scale feature fusion methods to obtain the graphs $\mathcal{G}_T = (\mathcal{A}_T, \mathcal{X})\}, \mathcal{G}_F = (\mathcal{A}_F, \mathcal{X})$ in feature space and topology space, respectively. We take the GCN-based encoder to learn the embeddings. We input $\mathcal{G}$ into the two-layer GCN as follows:

$$Z_1 = \mathrm{ReLU}\big(\hat{\mathcal{A}} X W^{(0)}\big) W^{(1)} \tag{6}$$

where Z_1 denotes the embedding result learned from the raw graph.

Similarly, $\mathcal{G}_F$ and $\mathcal{G}_T$ are input into the independent GCNs to obtain graph embeddings Z_2 and Z_3, respectively. For the three graph embeddings, namely Z_1, Z_2, and Z_3. we perform embedding fusion to generate Z

$$Z = \frac{1}{3}\sum_{i=1}^{3} Z_i \tag{7}$$

We assume that the subset $V_L = \{v_1, v_2, \dots, v_L\}$ of the node set V is the labeled training set, and the corresponding labels are$Y_L = \{y_1, y_2, \dots, y_L\}$. Additionally, V_U is the set of unlabeled nodes, and we can obtain the entire set $V = V_L \cup V_U$. For each node $v_l \in V_L$, there are two types of labels: the real label y_l and the predicted label Z_l. The cross-entropy error for node classification over all labeled nodes is then expressed as:

$$L_0 = -\sum_{v_l \in V_L} \sum_{t=1}^{C} y_l^t \ln Z_l^t \tag{8}$$

Here, C represents the total number of classes, y_l^t is the real label of node v_l for class t, and Z_l^t is the predicted probability for node v_l belonging to class t.

To ensure that $Z_2$and Z_3 remain close to Z_1 in the embedding space, we impose a consistency constraint that minimizes their deviation from the original graph embeddings. This ensures structural and feature similarity between the augmented and raw graphs. The constraint is defined as:

$$L_1 = \| Z_{2\text{nor}} - Z_{1\text{nor}} \|_F^2 + \| Z_{3\text{nor}} - Z_{1\text{nor}} \|_F^2 \tag{9}$$

Here, $Z_{1\text{nor}}$, $Z_{2\text{nor}}$ and $Z_{3\text{nor}}$ denote the L2-normalized embedding matrices of Z_1, Z_2 and Z_3,, respectively. By directly minimizing their distance, we enforce consistency and maintain stable representations.

3.3 Pseudo-Label Generation

Pseudo-labels play a crucial role in semi-supervised classification tasks, helping to mitigate the issue of label scarcity. However, existing pseudo-labeling methods often suffer from noisy labels and unreliable predictions, which can degrade model performance. To address this, we propose a confidence-based pseudo-label refinement approach that integrates confidence propagation and high-quality pseudo-label selection. Specifically, our method first propagates prediction confidence across the graph to enhance label consistency, then selects the most informative pseudo-labels based on an information gain criterion.

Given a graph $G = (V, E)$, where V is the set of nodes and E is the set of edges, each node $v \in V$ has an associated feature vector $x_v \in \mathbb{R}^d$. The goal is to classify nodes into C classes in a semi-supervised setting, where only a small subset of nodes has ground-truth labels.

Let $f_\theta(x_v)$ be a Multi-Layer Perceptron (MLP) that outputs the class probability distribution for node v :

$$y_{\text{pred},v} = \text{softmax}\big(f_\theta(x_v)\big) \in \mathbb{R}^C \tag{10}$$

where $y_{\text{pred},v}$ represents the predicted probability over C classes.

Since MLP operates independently on each node without considering graph structure, its predictions may be noisy or inconsistent. To address this, we introduce a confidence propagation mechanism followed by information gain-based pseudo-label selection.

To refine the confidence scores and leverage graph structure, we define an iterative confidence propagation process. Let $y_{\text{prop},v}^{(t)}$ be the propagated confidence of node v at iteration t. The update rule is:

$$y_{\text{prop},v}^{(t+1)} = \alpha \sum_{u \in N(v)} y_{\text{prop},u}^{(t)} + (1-\alpha) y_{\text{pred},v} \tag{11}$$

where $N(v)$ is the set of neighbors of node v, $\alpha \in (0,1)$ is a propagation factor controlling the influence of neighboring nodes, The sum $\sum_{u \in N(v)} \quad y_{\text{prop},u}^{(t)}$ aggregates confidence scores from neighbors.

This can be expressed in matrix form using the adjacency matrix A of the graph:

$$Y_{\text{prop}}^{(t+1)} = \alpha A Y_{\text{prop}}^{(t)} + (1-\alpha) Y_{\text{pred}} \tag{12}$$

where$Y_{\text{prop}} \in \mathbb{R}^{N \times C}$ is the confidence matrix for all N nodes.

The steady-state solution after sufficient iterations (assuming convergence) is:

$$Y_{\text{prop}} = (1-\alpha)(I - \alpha A)^{-1} Y_{\text{pred}} \tag{13}$$

where $(I - \alpha A)^{-1}$ can be interpreted as a smoothed label propagation operator. After propagation, Y_{prop} contains refined pseudo-label probabilities, making the predictions more consistent with the graph structure.

To prevent overly confident predictions, we apply row-wise normalization:

$$Y_{\text{prop}} = \frac{Y_{\text{prop}}}{\sum_c \ Y_{\text{prop},c}} \tag{14}$$

ensuring that the confidence distribution sums to 1 for each node.

After confidence propagation, we generate pseudo-labels by selecting the class with the highest probability:

$$\hat{y}_v = \arg max \ _c \, Y_{prop,v,c} \tag{15}$$

To filter out unreliable pseudo-labels, we compute the maximum confidence score for each node:

$$p_v = \max \, _c \, Y_{\text{prop},v,c} \tag{16}$$

A threshold τ is used to select only high-confidence pseudo-labels:

$$V_{\text{pseudo}} = \{ v \in V \mid p_v > \tau \} \tag{17}$$

where V_{pseudo} is the set of pseudo-labeled nodes. However, this selection does not consider graph consistency, which we address next.

Since individual node confidence may still be misleading, we introduce an information gain (IG) score to evaluate the consistency of a pseudo-label with its local neighborhood.

For each node v, we define its information gain score as the average confidence of its neighbors for the same predicted class:

$$\mathrm{IG}(v) = \frac{1}{|\,N(v)\,|} \sum_{u \in N(v)} Y_{\mathrm{prop},u,\hat{y}_v} \tag{18}$$

This score quantifies how strongly node v's pseudo-label aligns with its neighborhood. A higher score means the node and its neighbors consistently predict the same class with high confidence.

To select high-quality pseudo-labels, we take the top k nodes per class with the highest IG scores:

$$V_{\mathrm{selected}}(c) = \arg\max_{\,v \in V_{\mathrm{pseudo}}, \hat{y}_v = c} \mathrm{IG}(v)\text{, for each class } c \tag{19}$$

where$V_{\mathrm{selected}}(c)$ contains the top k pseudo-labeled nodes for class c. The final refined pseudo-labels are:

$$\hat{y}_v = \begin{cases} \hat{y}_v, & \text{if} v \in V_{\mathrm{selectesd}} \\ \text{unlabeled}, & \text{otherwise} \end{cases} \tag{20}$$

We define Q_v as a one-hot encoded distribution corresponding to the refined pseudo-label $\hat{y}_v$:

$$Q_v(c) = \begin{cases} 1, & \text{if } c = \hat{y}_v, \\ 0, & \text{otherwise.} \end{cases} \tag{21}$$

The KL divergence between Q_v and P_v for node v is given by:

$$\mathrm{KL}(Q_v \parallel P_v) = \sum_{c=1}^{C} Q_v(c) \log \frac{Q_v(c)}{P_v(c)}. \tag{22}$$

Since Q_v is a one-hot distribution, only the term corresponding to $c = \hat{y}_v$ is non-zero, and the expression simplifies to:

$$\mathrm{KL}(Q_v \parallel P_v) = -\log P_v(\hat{y}_v). \tag{23}$$

Thus, the overall pseudo-label loss over all selected nodes V_{selected} becomes:

$$\mathcal{L}_{\mathrm{pseudo}} = \sum_{v \in V_{\mathrm{selected}}} -\log P_v(\hat{y}_v), \tag{24}$$

where $P_v(\hat{y}_v)$ denotes the predicted probability of node v for the refined pseudo-label $\hat{y}_v$.

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

$$L = L_0 + L_1 + \lambda \mathcal{L}_{\mathrm{pseudo}} \tag{25}$$

where λ is a balance hyper-parameter.

4 EXPERIMENTS

In this section, we compare our model with several popular graph-based semi-supervised node classification models. Subsequently, we conduct ablation experiments to evaluate the performance of each component of MCF-GCN. The model accuracy and its visualization further demonstrate the effectiveness of MCF-GCN.

4.1 Datasets and Experimental Setup

Our experiments are based on three citation graphs: Cora, CiteSeer, and PubMed [20], with nodes indicating papers and edges indicating citations. Node features are generated using a bag-of-words representation, and each node has a class label. For node classification, we use the default dataset split: 20 instances for training, 500 for validation, and 1000 for testing, with the remaining labels unused [21]. Statistics for these datasets are summarized in **Table 1**.

Table 1: Dataset statistics

Dataset	Classes	Nodes	Edges	Features
Cora	7	2708	5429	1433
Citeseer	6	3327	4732	3703
PubMed	3	19717	443328	500

The learning rate is set to 0.01, and the Adam optimizer [19] is used, with a hidden layer dimension of $\{32,64,128\}$. The dropout rate for the adjacency matrix is 0.5, weight decay is set to 5e-4. A two-layer Graph Convolutional Network is applied to each dataset. In our experiment, m and n are set to the same value, with both parameters ranging from $\{1,2,3,4,5,6\}$. A sensitivity analysis is conducted to evaluate their impact.

In addition, The selection of pseudo-labels involves three key parameters: the smoothing factor α, the pseudo-label confidence threshold τ, and the maximum number of pseudo-labels per class k. The smoothing factor α controls the influence of neighboring nodes during confidence propagation and is chosen from $\{0.5, 0.6, 0.7, 0.8, 0.9\}$. The pseudo-label confidence threshold τ determines the minimum confidence required for a node to be considered as a pseudo-label, with values selected from $\{0.7, 0.8, 0.9, 0.95\}$. The maximum number of pseudo-labels per class k restricts the number of selected pseudo-labels based on information gain and is chosen within the range of [30 - 100].

4.2 Baselines

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

- Base encoder: GCN [3], SGC [8], GAT [6].
- Sampling-based encoder: FastGCN [22].

- Multi-scale information fusion-based encoder: N-GCN[13].
- Graph generation fusion-based encoder: MOGCN[23], PA-GCN[24].
- Pseudo labeling-based encoder: MFGCN[18].

4.3 Experiment Results

Node Classification

We use the proposed overall architecture to perform node classification tasks on the Cora, CiteSeer, and PubMed citation network datasets.We report the average accuracy when using a training set of 20 nodes per class and compare it with other methods. The results are shown in **Table 2**.

Table 2: Node classification results

Model	Dataset		
	Cora	**Citeseer**	**PubMed**
GCN	81.51	70.72	78.80
GAT	82.48	72.08	79.08
SGC	81.90	72.21	78.30
FastGCN	80.70	70.70	79.40
N-GCN	83.00	72.20	79.50
MOGCN	83.10	72.40	79.20
PA-GCN	83.60	70.40	79.30
MFGCN	75.10	71.90	79.10
Ours	**85.60**	**74.10**	**81.00**

MCF-GCN outperforms the multi-scale information fusion-based encoders (MOGCN, N-GCN, PA-GCN), multi-view fusion-based encoder (MFGCN), and sampling-based encoder (FastGCN) on the three datasets.

Ablation Experiment.

To clearly demonstrate the effectiveness of the two modules in the model, we conducted ablation experiments to highlight the contribution of each component. The design of the two ablation experiments is as follows, with the results selecting the highest average accuracy, as shown in **Table 3**.

1. GCN #A: GCN \#A refers to a variant of GCN,which serves as the basic GCN integrated with the graph generation and multi-channel fusion modules.
2. GCN #B: GCN \#B refers to a variant of GCN, which serves as the basic GCN integrated with the pseudo-label generation module.

For GCN # A, combining the multi-channel fusion module with GCN after graph generation results in a slight improvement in node classification accuracy: 1.1%, 2.1%, and 0.4% on the Cora, Citeseer, and PubMed datasets, respectively, compared to the baseline.

Table 3: Accuracy(%) of ablation experiments on three datasets

Model	Dataset		
	Cora	Citeseer	PubMed
GCN	81.5	70.9	79.4
GCN #A	82.6	73.0	79.8
GCN #B	83.8	73.4	80.3

For GCN # B, combining the pseudo-label generation module with GCN significantly improves node classification, achieving increases of 2.3%, 2.5%, and 0.9% on the Cora, Citeseer, and PubMed datasets, respectively, compared to the GCN baseline. Meanwhile, we compare our method with other traditional pseudo-label-based node classification approaches, with the results presented in **Table 4**. This demonstrates that our pseudo-labeling module effectively addresses the label sparsity issue in semi-supervised node classification.

Table 4: Accuracy(%) of Pseudo-label generation methods on three datasets

Model	Dataset		
	Cora	Citeseer	PubMed
GCN-LP[25]	82.6	72.6	79.6
Self-training[26]	80.8	71.0	77.0
GCN-Feature-sim	82.5	71.5	79.0
GCN #B	83.8	73.4	80.3

To evaluate whether the pseudo-labels generated by our method are of higher quality than those produced by other approaches, we analyze the neighborhood consistency of pseudo-labels generated by different methods, as shown in **Fig. 2**. The results demonstrate that our method produces higher-quality pseudo-labels with better neighborhood consistency compared to other approaches.

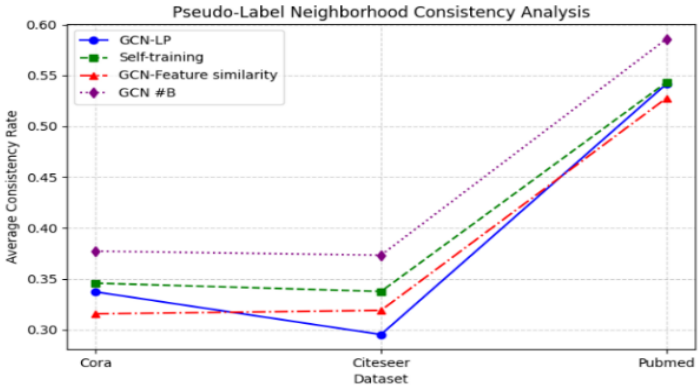
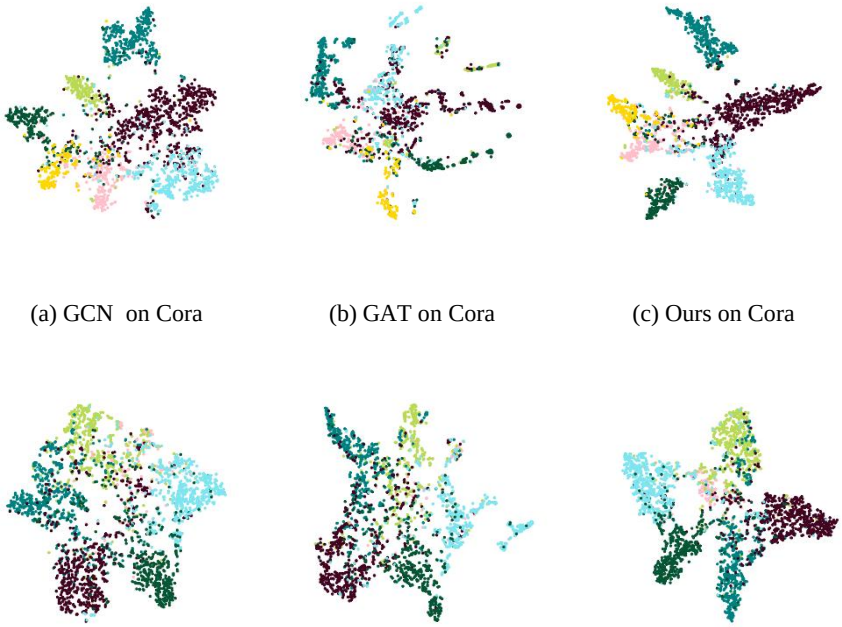


Fig. 2: Pseudo-Label Neighborhood Consistency Analysis of Four Pseudo-Label Generation Methods

Classification Result Visualization.
In order to more intuitively demonstrate the effectiveness of our proposed model, we perform a visualization task (distribution of raw data and classification results of GCN, GAT, and MCF-GCN) on the Cora and Citeseer datasets using the t-SNE [27] algorithm. As shown in **Fig. 3**, the visualization results of GCN and GAT are unsatisfactory, as there is no clear boundary between different classes. In contrast, the proposed model performs better, with a larger inter-class distance and smaller intra-class distance.



(a) GCN on Cora (b) GAT on Cora (c) Ours on Cora

(d) GCN on Citeseer (e) GAT on Citeseer (f) Ours on Citeseer

Fig. 3: The visualization of classification results of GCN, GAT, and Our method on Cora, Citeseer datasets.

5 Conclusion

This paper proposes an end-to-end multi-channel Graph Convolutional Network (MCF-GCN) model for semi-supervised node classification. Our model constructs multiple graphs by processing the original topology and node features from different perspectives. A multi-channel GCN encoder is then employed to extract features from these graphs and integrate them, enabling a more comprehensive representation of node information. Furthermore, we introduce a pseudo-label generation mechanism based on confidence propagation and information gain filtering, which effectively selects high-

quality pseudo-labels to mitigate the issue of label sparsity. Experimental results on three public datasets demonstrate that our model outperforms existing methods, validating its effectiveness in semi-supervised node classification.

References

1. Chien, E., Peng, J., Li, P., Milenkovic, O.: Adaptive universal generalized pagerank graph neural network. arXiv preprint arXiv:2006.07988. (2020).
2. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. Advances in neural information processing systems. 30, (2017).
3. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907. (2016).
4. Liang, H., Du, X., Zhu, B., Ma, Z., Chen, K., Gao, J.: Graph contrastive learning with implicit augmentations. Neural Networks. 163, 156–164 (2023).
5. Sun, K., Lin, Z., Zhu, Z.: Multi-stage self-supervised learning for graph convolutional networks on graphs with few labeled nodes. In: Proceedings of the AAAI conference on artificial intelligence. pp. 5892–5899 (2020).
6. Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. stat. 1050, 10–48550 (2017).
7. Wang, X., Yang, X., Wang, P., Yu, H., Xu, T.: SSGCN: a sampling sequential guided graph convolutional network. International Journal of Machine Learning and Cybernetics. 15, 2023–2038 (2024).
8. Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., Weinberger, K.: Simplifying graph convolutional networks. In: International conference on machine learning. pp. 6861–6871. Pmlr (2019).
9. Dong, H., Li, L., Tian, D., Sun, Y., Zhao, Y.: Dynamic link prediction by learning the representation of node-pair via graph neural networks. Expert Systems with Applications. 241, 122685 (2024).
10. Zhang, P., Chen, J., Che, C., Zhang, L., Jin, B., Zhu, Y.: IEA-GNN: Anchor-aware graph neural network fused with information entropy for node classification and link prediction. Information Sciences. 634, 665–676 (2023).
11. Ma, T., Wang, H., Zhang, L., Tian, Y., Al-Nabhan, N.: Graph classification based on structural features of significant nodes and spatial convolutional neural networks. Neurocomputing. 423, 639–650 (2021).
12. Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O., Dahl, G.E.: Neural message passing for quantum chemistry. In: International conference on machine learning. pp. 1263–1272. PMLR (2017).
13. Abu-El-Haija, S., Kapoor, A., Perozzi, B., Lee, J.: N-gcn: Multi-scale graph convolution for semi-supervised node classification. In: uncertainty in artificial intelligence. pp. 841–851. PMLR (2020).
14. Shen, X., Sun, D., Pan, S., Zhou, X., Yang, L.T.: Neighbor contrastive learning on learnable graph augmentation. In: Proceedings of the AAAI conference on artificial intelligence. pp. 9782–9791 (2023).
15. Yao, K., Liang, J., Liang, J., Li, M., Cao, F.: Multi-view graph convolutional networks with attention mechanism. Artificial Intelligence. 307, 103708 (2022).
16. Ziraki, N., Dornaika, F., Bosaghzadeh, A.: Multiple-view flexible semi-supervised classification through consistent graph construction and label propagation. Neural Networks. 146, 174–180 (2022).

17. Wang, X., Zhu, M., Bo, D., Cui, P., Shi, C., Pei, J.: Am-gcn: Adaptive multi-channel graph convolutional networks. In: Proceedings of the 26th ACM SIGKDD International conference on knowledge discovery & data mining. pp. 1243–1253 (2020).
18. Yang, Y., Sun, Y., Ju, F., Wang, S., Gao, J., Yin, B.: Multi-graph fusion graph convolutional networks with pseudo-label supervision. Neural Networks. 158, 305–317 (2023).
19. Kinga, D., Adam, J.B.: A method for stochastic optimization. In: International conference on learning representations (ICLR). San Diego, California; (2015).
20. Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., Eliassi-Rad, T.: Collective classification in network data. AI magazine. 29, 93–93 (2008).
21. Yang, Z., Cohen, W., Salakhudinov, R.: Revisiting semi-supervised learning with graph embeddings. In: International conference on machine learning. pp. 40–48. PMLR (2016).
22. Chen, J., Ma, T., Xiao, C.: Fastgcn: fast learning with graph convolutional networks via importance sampling. arXiv preprint arXiv:1801.10247. (2018).
23. Li, X., Ma, J., Leng, L., Han, M., Li, M., He, F., Zhu, Y.: MoGCN: a multi-omics integration method based on graph convolutional network for cancer subtype analysis. Frontiers in Genetics. 13, 806842 (2022).
24. Guo, Q., Yang, X., Zhang, F., Xu, T.: Perturbation-augmented graph convolutional networks: A graph contrastive learning architecture for effective node classification tasks. Engineering Applications of Artificial Intelligence. 129, 107616 (2024).
25. Wang, H., Leskovec, J.: Unifying graph convolutional neural networks and label propagation. arXiv preprint arXiv:2002.06755. (2020).
26. Li, Q., Han, Z., Wu, X.-M.: Deeper insights into graph convolutional networks for semi-supervised learning. In: Proceedings of the AAAI conference on artificial intelligence (2018).
27. Van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. Journal of machine learning research. 9, (2008).