Individuality-enhanced and multi-granularity consistency-preserving graph neural network for semi-supervised node classification

Xinxin Liu[1] · Weiren Yu[2]

Abstract

Semi-supervised node classification is an important task that aims at classifying nodes based on the graph structure, node features, and class labels for a subset of nodes. While most graph convolutional networks (GCNs) perform well when an ample number of labeled nodes are available, they often degenerate when the amount of labeled data is limited. To address this problem, we propose a scheme, namely, Individuality-enhanced and Multi-granularity Consistency-preserving graph neural Network (IMCN), which can alleviate the problem of losing individual information within the encoder while providing a reliable supervised signal for learning purposes. First, one simple encoder based on node features only is integrated to enhance node individuality and amend node commonality learned by the GCN-based encoder. Then, three constraints are defined at different levels of granularity, encompassing node embedding agreement, semantic class alignment, and node-to-class distribution identity. They can maintain the consistency between the individuality and commonality of nodes and be leveraged as latent supervised signals for learning representative embeddings. Finally, the trade-off between the individuality and commonality of nodes captured by two encoders is taken into consideration for node classification. Extensive experiments on six real-world datasets have been conducted to validate the superiority of IMCN against state-of-the-art baselines in handling node classification tasks with scarce labeled data.

1 Introduction

Graphs are useful data structures for capturing relationships between entities in various complex interconnected systems, such as social relationships [1], protein interactions [2], commodity co-purchasing [3], and co-citations [4]. Many fundamental tasks on graphs involve making predictions over nodes, such as predicting labels for unlabeled nodes according to the graph structure and node attributes. In practical scenarios, the availability of labeled nodes is often limited due to the resource-intensive and time-consuming nature of the annotation process [5, 6]. Consequently, the scarcity of labeled examples renders the task of classifying nodes in a semi-supervised manner both challenging and crucial.

In recent years, deep graph neural networks [7–10] have garnered considerable attention and witnessed notable advancements in semi-supervised node classification. Among them, graph convolutional networks (GCNs) have emerged as a prominent approach. Early GCNs, such as GCN [11] and the hierarchical GCN model [12], are based on the graph convolution process, which focuses on propagating and aggregating feature information between adjacent nodes. These models solely learn node embeddings from the single-view perspective, often adopting a shallow architecture to mitigate the problem of over-smoothing [13–15] that may arise when dealing with node feature information. These characters often lead to the problem of capturing only a restricted amount of information by most single-view GCNs. This is unfavorable for learning discriminative node embeddings and devising effective classifiers, particularly in situations where the availability of labeled nodes is scarce in practical tasks.

✉ Weiren Yu
weiren.yu@warwick.ac.uk

Xinxin Liu
xinxinliucn@njust.edu.cn

[1] School of Computer Science and Technology, Nanjing University of Science and Technology, Jiangsu, China

[2] Department of Computer Science, University of Warwick, Coventry, UK

Inspired by the recent success of contrastive learning [16–18] in computer vision, numerous multi-view GCNs have been specifically designed to capture node feature information from diverse perspectives. There are two primary approaches for generating node embeddings from multi-views: one is combining different kinds of information from the same graph, such as the models CG^3 [19] and VCHN [20]; and the other is contrasting the information captured from one graph and its corresponding augmented one, such as the models MVGRL [21] and CSGNN [22]. Therefore, information from different views can be complementary by each other and provide valuable hints for promoting node embedding and classification.

Despite the remarkable advancements achieved by single-view and multi-view GCNs in semi-supervised node classification, it remains a daunting challenge to effectively classify nodes with only a limited number of labeled examples (lack of supervision). This is mainly attributed to the following two problems: (1) Graph convolutions focus on information propagations from neighboring nodes to the central node based on the graph topology, so that the similar features or commonality shared by connected nodes are learned. However, this may result in a loss of the node individuality, the acquisition of over-smoothed features, and the failure to distinguish nodes from different classes [13]. Such outcomes are unfavorable when aiming to learn a discriminative classifier. (2) The class label information of nodes is scarce and concise but plays a significant role in supervising model learning. The relation between nodes and classes should be taken full advantage of for mining valuable supervised signals and optimizing the processes of node embedding and classification. However, only limited useful supervised signals are mined from the relation by most GCNs [19, 20].

In this paper, we develop an Individuality-enhanced and Multi-granularity Consistency-preserving graph neural Network (IMCN) for semi-supervised node classification with scarce labeled data. Specifically, IMCN integrates a simple two-layer MLP as a supplementary encoder to amend the individuality of nodes damaged by graph convolutions. Then, IMCN enriches supervised signals by taking full advantage of the multi-granularity relations among nodes and classes. The main contributions are summarized as follows:

1) An individuality-enhanced and multi-granularity consistency-preserving graph neural Network is built for semi-supervised node classification, which can maintain the individuality and commonality of nodes simultaneously during the feature extraction process. The proposed method is highly effective, particularly when there are only a few labeled nodes available for model training.

2) Three consistency constraints at different granularity are designed to enrich the supervised information for model learning: the fine-grained one at the node level by an improved semi-supervised contrastive loss; the coarse-grained one at the semantic class level by aligning the prototypes of the same class learned from different encoders; and the middle-grained one at the node-to-class level by ensuring the identity of node-to-class relational distributions learned from two encoders.

3) Extensive experiments on six real-world networks from different fields verify that IMCN significantly outperforms the comparison methods for semi-supervised node classification with few labeled nodes. Especially, on the three public benchmark datasets Cora, CiteSeer, and PubMed, the classification accuracies of IMCN are more than 2.5% higher than baseline methods when only two or three labeled nodes per class are available for model training.

The remainder of the paper is structured as follows. Section 2 introduces some related work. Section 3 presents the details of the proposed model. Then, the experimental setup and results are introduced and analyzed in Section 4 and Section 5, respectively. Finally, Section 6 provides the main conclusions of this paper and gives ideas for further study.

2 Related work

In this section, some previous work on GCNs for semi-supervised node classification are briefly reviewed based on whether the method capture abundant information from different aspects.

2.1 Single-view GCNs

Single-view GCNs usually learn node embeddings for classification by propagating and aggregating feature information between adjacent nodes in the graph from one aspect only. The classical and representative model GCN [11], which derived inspiration primarily from the convolutional operations on images, learns low-dimensional node embeddings through propagations and aggregations of nodes' and their neighbors' features. The GraphSAGE (SAmple and aggreGatE) model [23], a general inductive framework, generates node embeddings by sampling and aggregating features from one node's local neighborhood and can efficiently generate node embedding for previously unseen data. The Graph ATtention network (GAT) [24] employs an attention mechanism to modify the traditional propagation and aggregation operations between one node and its neighbors in GCN. The Simple Graph Convolution network (SGC) [25] reduces the excess complexity in GCN by successively removing nonlinearities and collapsing weight matrices between consecutive layers. The Hierarchical Graph Convolution Network (H-GCN) [12] enlarges the receptive field of graph convolutional processes in GCN by fusing nodes with similar structures into super-nodes. The simplified multi-layer graph convolutional

networks with dropout [6] combines SGC and the dropout regularization in deep learning and extends the shallow GCN model to the multi-layer GCN model to extract information from higher-order neighbors. This model can reduce redundant calculation and over-fitting of the multi-layer GCN to make it simple and efficient.

Despite the noticeable achievements of these single-view models, the main concern of lacking supervised information in semi-supervised node classification is still not well handled. Most single-view GCNs are usually shallow with one aspect only taken into consideration, and cannot obtain adequate information for effective classification when very limited nodes are labeled.

2.2 Multi-view GCNs

Different from the single-view GCNs, multi-view methods are specifically designed to capture abundant information from different aspects for improving learning and classification. In recent years, many multi-view GCNs have been proposed, and are classified into the following two classes.

One captures and combines feature information from two views of the same graph. The Contrastive GCN with Graph Generation (CG^3) [19] integrates H-GCN and a two-layer GCN to learn complementary information from local and global views of nodes and imposes the designed node-level contrastive and graph-level generative constraints on the embeddings learned by the above two encoders. The View-Consistent Heterogeneous Network (VCHN) [20] combines the classical methods GCN and GAT to learn node embeddings from spectral and spatial views and applies constraints on the predictions between two views to promote the supervision from one to the other.

The other captures feature information from one graph and its corresponding augmented graph and contrasts them to provide extra useful information for learning. The Deep Graph Infomax model (DGI) [26] aims at learning patch representations and the corresponding high-level summaries of graphs and related corrupted graphs by GCN, and then maximizes the mutual information between them. Similar to DGI, the contrastive Multi-View Graph Representation Learning model (MVGRL) [21] learns node representations for the graph and its corresponding corrupted one by two different GNNs and a shared MLP and generates corresponding graph representations from them by shared pooling and MLP layers. Then, contrastive constraints between node and graph representations are designed as important parts of the learning objective. The deep GRAph Contrastive rEpresentation learning model (GRACE) [27] first generates two correlated graph views by randomly performing corruption of removing edges and masking node features, then maximizes the agreement between node embeddings in these two views based on the idea of contrastive learning. The Contrastive Semi-supervised learning model based GNN (CSGNN) [22] employs a two-layer GCN as a teacher encoder to learn node representations for one graph and its corrupted graph, and then contrasts the latent vectors between nodes, edges, and labels from these two views for improving predictions. In the final stage, the predictions are distilled into the downstreaming student module.

The above two branches of multi-view GCNs can learn complementary information for boosting the discrimination of node embeddings and classification accuracy. However, these methods usually rely on the graph convolutional process, which enforces the encoders focusing on the commonality of adjacent nodes and damaging some individuality of them. In addition, these models do not take full advantage of the complex but valuable relation information among nodes and classes.

The methods H-GCN, CG^3, and VCHN are closely related to the method proposed in this paper. The constraints and mechanism used in these four methods are listed in Table 1.

The following are the differences among the proposed IMCN method, H-GCN, CG^3, and VCHN. Compared with the multi-view methods CG^3 and VCHN, a GCN-based encoder is coupled with an MLP-based encoder in the proposed IMCN method, which can enhance the node individuality for learning discriminative node embedding vectors. For the node-level constraints in IMCN and CG^3, the calculation of the former is much simpler than that of the latter by reducing repeated node-pair contrasts. In addition, IMCN takes full advantage of the complex relations among nodes and classes from the views of node-to-class distribution and class centroid alignment which are ignored by the other three methods.

Table 1 Differences among H-GCN, CG^3, VCHN, and the proposed IMCN method

Constraints/Mechanism	H-GCN	CG^3	VCHN	IMCN
multi-view encoder	—	√	√	Individuality-enhanced
node-level constraint	—	√	√	No repeated calculation
node2class-level constraint	—	—	—	√
class-level constraint	—	—	—	√

3 IMCN model

This section first introduces the framework of the proposed Individuality-enhanced and Multi-granularity Consistency-preserving graph neural Network (IMCN) for semi-supervised node classification. Then, the critical components of IMCN are described in detail.

3.1 Framework overview

We design a novel graph neural network (as shown in Fig. 1) for semi-supervised node classification, which commits to taking both the individual- and common-feature information of nodes and preserving multi-granularity consistency between them to learn discriminative node representations for effective classification. The model learning and classification process of IMCN mainly includes the following three stages:

1) **Feature extraction** The individual- and common-feature information for nodes is extracted by two different encoders according to the graph topology and original node features.

2) **Multi-granularity consistency constrains** The multi-granularity consistency of feature information learned from the encoders is preserved according to the relations among nodes and classes.

3) **Feature fusion and node classification** A trade-off is made between the constrained individuality and commonality of nodes and then a multi-objective loss function is established to obtain the optimized model for node classification.

3.2 Feature extraction

Different views contain quite different information to describe the same object, which can provide complementary infor-

mation to improve model learning. Multi-view learning has grown in popularity as a result of this concept. Meanwhile, contrastive learning has received intensive research in recent years showing that contrasting congruent and incongruent views of objects can help the algorithms learn expressive representations [28–32]. Inspired by these ideas, two different views are established for a graph and applied to learn discriminative node representations for classification.

Although some augmentation strategies are proposed to generate related graphs with different views, such as node dropping and edge perturbation in [21, 22, 33], they may destroy the original graph topology and degrade the performance of graph convolutional networks. Unlike the previous approaches, in this paper, the node itself is seen as a local view and the nodes with adjacent neighbor structures as a global view for a graph. These two views are obviously different from those of node2vec [34] which takes the width and depth of the deep walks controlled by two parameters as local and global views.

From the global structure view, many public GCN-based models can be used to capture common-feature information among adjacent nodes. Here, the effective GCN-based model H-GCN [12] is adopted as the global encoder in the proposed IMCN model. H-GCN aggregates nodes that have equivalent or similar structures into hyper-nodes for graph convolution and then refines the coarsened graphs to the original graphs for restoring the representation for each node. Therefore, the receptive field for each node is enlarged, and more global and common-feature information of nodes can be comprehensively captured. The node feature matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$ and adjacency matrix $\mathbf{A} \in \mathbb{R}^{m \times m}$ are input into the H-GCN encoder to generate low-dimension global node representations $\mathbf{H}^{global} \in \mathbb{R}^{m \times c}$ as follows:

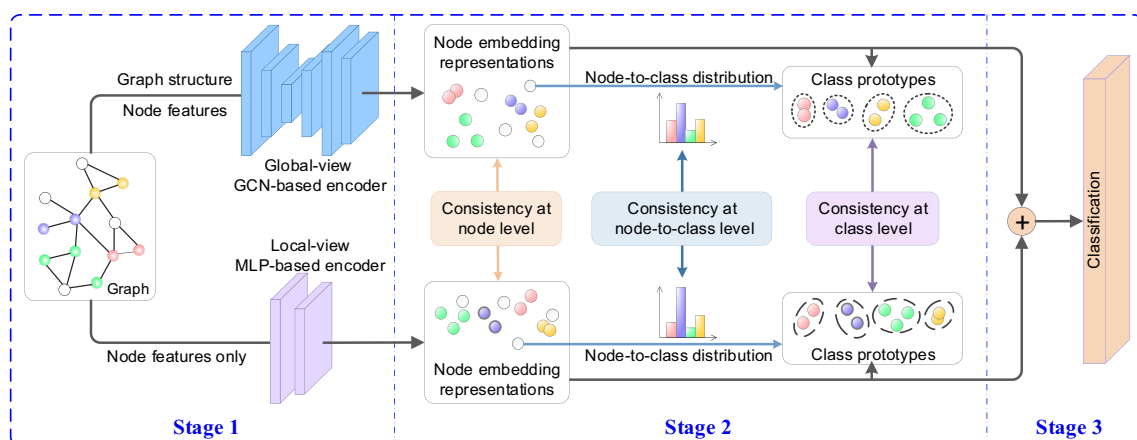$$\mathbf{H}^{global} = \phi(\mathbf{A}, \mathbf{X}), \tag{1}$$

Fig. 1 Framework of the proposed IMCN model: feature extraction in Stage 1; multi-granularity consistency constrains in Stage 2; and feature fusion and node classification in Stage 3

where m, n, and c (as the dimension of node embeddings) are the number of nodes, node features, and classes, respectively; and $\phi(\cdot)$ denotes the processes of generating coarse graphs, graph convolution, and refining coarsened graphs in H-GCN.

However, those GCN-based encoders mainly focus on the commonality of linked nodes excessively and may lose the individuality of nodes in information propagation. This problem also exists in the local encoder which is designed by two-layer GCNs in [19]. In practice, the categories of nodes are mainly determined by their individual-feature information. Therefore, to compensate for the damaged individual-feature information learned from the global GCN-based encoder, the node itself is regarded as a local view and extract the individual-feature information of nodes by a simple two-layer MLP encoder with $\mathbf{X}$ as the only input. The local node low-dimension representations $\mathbf{H}^{local} \in \mathbb{R}^{m \times c}$ can be obtained by IMCN as follows:

$$\mathbf{H}^{local} = \sigma(\mathbf{X}\mathbf{W}^{(0)})\mathbf{W}^{(1)}, \tag{2}$$

where $\mathbf{W}^{(i)}$ and $\sigma(\cdot)$ denote the trainable weight matrix and non-linear ReLU activation function [35]. Denote $\mathbf{W}^{(0)} \in \mathbb{R}^{n \times d}$, and $\mathbf{W}^{(1)} \in \mathbb{R}^{d \times c}$, where d is the feature dimension in the hidden layer. Since $\mathbf{H}^{local}$ is computed regardless of the structural information of graphs, the label information can be effectively propagated without the limitation of distance between nodes.

In order to make the feature information learned by two encoders in the same metric, $\mathbf{H}^{local}$ and $\mathbf{H}^{global}$ are normalized by the L_2-norm in the column direction before imposing the following multi-granularity consistency constraints and the classification stage.

3.3 Multi-granularity consistency constraints

For the individuality and commonality of nodes captured separately through the above feature extraction processes, it is reasonable to preserve the consistency between them for optimizing the encoding process. Inspired by human cognition and intelligence, data are analyzed from different granularities in IMCN. This strategy can lead the model to analyze data more comprehensively, utilize data more efficiently, and make more accurate decisions. The following part introduces the designed multi-granularity consistency constraints according to relations among nodes and classes in detail.

3.3.1 Node-level consistency constraint

Data tend to be analyzed intuitively from the node (sample) level, which can enforce the model focusing on the features of representative samples and having good generalization ability. From this level, some common information shared by

local and global representations of one node is described as fine-grained node-level consistency.

In the proposed IMCN method, the vector distance of local and global representations is used to measure this fine-grained consistency. In detail, this constraint is defined with unsupervised and supervised parts as follows.

On the one hand, in order to utilize the abundance of unlabeled information effectively, an unsupervised node-level loss is defined to maintain the consistency between the local and global representations of the same node:

$$L_{node}^u = -\log \frac{\sum_{i=1}^m e^{sim(\mathbf{h}_i^{local}, \mathbf{h}_i^{global})}}{\sum_{j,k=1}^n e^{sim(\mathbf{h}_j^{local}, \mathbf{h}_k^{global})}}, \tag{3}$$

where $\mathbf{h}_i^{local}$ is the i-th row vector in $\mathbf{H}^{local}$, and $sim(\mathbf{a}, \mathbf{b})$ is the cosine similarity between $\mathbf{a}$ and $\mathbf{b}$: $sim(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{|\mathbf{a}| \times |\mathbf{b}|}$. By minimizing this loss, the representations of the same node from two views are expected to be similar, while those of different nodes are expected to be away from each other.

On the other hand, labeled nodes are scarce but can provide valuable semantic information for learning expressive node embeddings for easy classification. The consistency between local and global representations of labeled nodes belonging to the same class is maintained by a designed supervised loss as follows:

$$L_{node}^s = -\log \frac{\sum_{i,j=1;\mathbf{y}_i=\mathbf{y}_j}^m e^{sim(\mathbf{h}_i^{local}, \mathbf{h}_j^{global})}}{\sum_{k,m=1}^n e^{sim(\mathbf{h}_k^{local}, \mathbf{h}_m^{global})}}, \tag{4}$$

where $\mathbf{y}_i$ is the one-hot coded class vector of the i-th node and $\mathbf{y}_i \in \mathbb{R}^{1 \times c}$. Therefore, each labeled node from one view is contrasted with the labeled nodes belonging to the same class from the other view.

Note that in the above two loss terms, we expect node embeddings of the same node or nodes from the same class to be most similar simultaneously in the joint similarity distribution between all nodes, instead of the marginal distribution, as shown in Fig. 2. This is more reasonable and can avoid the following time-consuming duplication calculations in CG^3 [19]: (1) reuse negative nodes (blue shaded and checked ones as shown in Fig. 2(a) and (c)) in contrastive learning; and (2) repeat calculations of the inner products between row vectors from $\mathbf{H}^{local}$ and $\mathbf{H}^{global}$ in each loss term.

Finally, the local-global consistency at the fine-grained node level is maintained by a node-wise regularization term defined with both the unsupervised and supervised information as the following formula:

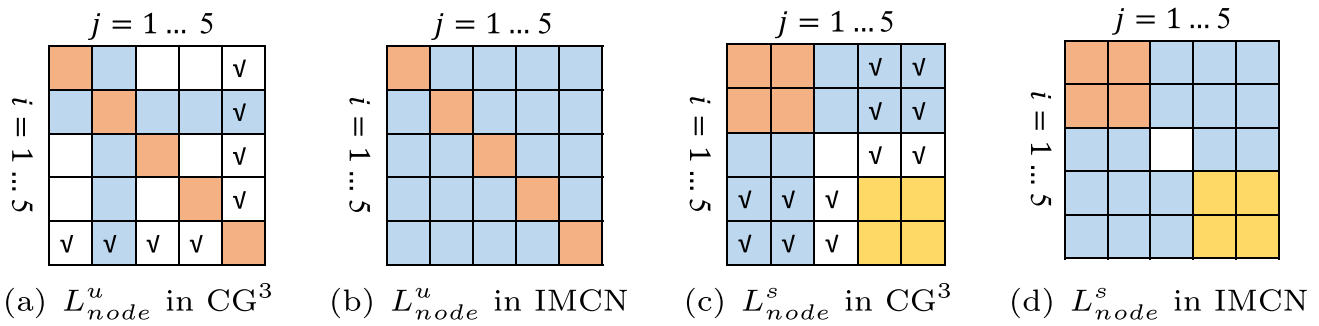$$L_{node} = L_{node}^s + L_{node}^u. \tag{5}$$

Fig. 2 Differences between the proposed IMCN method and CG³ [19] on calculating unsupervised and supervised node-level loss

Therefore, the learning process of IMCN for local and global node representations can complement and promote each other based on node features and semantic information.

3.3.2 Class-level consistency constraint

The performance of the model tends to be biased when trained only at the sample level. The reason is that different samples can share some common features and belong to the same class, and it is necessary for the model to distinguish between samples of different classes. Despite a small number of labeled nodes, their semantic category information is an important supplement for feature embedding. This is not taken into consideration in [19]. From the perspective of semantic category, the common information between the local and global views of nodes is named after coarse-grained class-level consistency.

Following but different from the idea in [36], prototypes for each class from local and global views are generated using the learned embeddings of the labeled nodes, then the distance between them are expected to be minimized. The following constraint is designed:

$$L_{class} = \frac{1}{c} \sum_{i=1}^{c} \left\| \mathbf{c}_i^{local} - \mathbf{c}_i^{global} \right\|_2^2, \tag{6}$$

where $\mathbf{c}_i^{local} \in \mathbb{R}^c$ and $\mathbf{c}_i^{global} \in \mathbb{R}^c$ are the prototypes of the i-th class calculated by average aggregation of the learned local and global embeddings of the labeled nodes belonging to this class respectively, $\|\cdot\|_2$ is the L_2-norm operator, and L_{class} is the mean-squared Euclidean distance of the corresponding class prototypes. Note that it is different from the magnet loss [37] which uses the k-means method to compute cluster centers for each class.

The representations of class prototypes are not stable during the model learning process and may forget valuable information learned before. Therefore, in the t-th iteration, we compute the class prototypes $\mathbf{c}_i^{local}$ and $\mathbf{c}_i^{global}$ in the way

mentioned above, then add them to the prototype representations calculated after the last iteration for updating class prototype representations and suppressing the instability:

$$\mathbf{c}_i^{local(t)} = (1 - \mu)\mathbf{c}_i^{local(t-1)} + \mu\mathbf{c}_i^{local},$$
$$\mathbf{c}_i^{global(t)} = (1 - \mu)\mathbf{c}_i^{global(t-1)} + \mu\mathbf{c}_i^{global}, \tag{7}$$

where μ is the balance weight for updating the class prototypes in the t-th iteration based on the prototype representation after $t - 1$ iterations, and $\mu \in [0, 1)$.

3.3.3 Consistency constraint at the node-to-class level

Assume that the distribution around each prototype is isotropic Gaussian and that the distributions around the same class in the local and global views should be similar. Therefore, in addition to the consistencies at node and class levels, there must be some indispensable consistent information in the node-to-class relationship between local and global views. This is also not taken into consideration in [19].

To make the best use of unlabeled nodes, $sim(\mathbf{h}_i, \mathbf{c}_j)$ is used to calculate the similarities between each node embedding and the obtained class prototypes, and then node-to-class relational distributions are generated for unlabeled nodes in local and global views according to the following expressions:

$$p_{ij}^{local} = \frac{e^{sim(\mathbf{h}_i^{local}, \mathbf{c}_j^{local})/\tau}}{\sum_{k=1}^{c} e^{sim(\mathbf{h}_i^{local}, \mathbf{c}_k^{local})/\tau}},$$
$$p_{ij}^{global} = \frac{e^{sim(\mathbf{h}_i^{global}, \mathbf{c}_j^{global})/\tau}}{\sum_{k=1}^{c} e^{sim(\mathbf{h}_i^{global}, \mathbf{c}_k^{global})/\tau}}, \tag{8}$$

where $\tau > 0$ is a temperature hyper-parameter denoting the concentration of node embeddings around class prototypes, and a smaller τ indicates a larger concentration.

Then, the distribution of the relation of one node to all classes can be represented as $\mathbf{p}_i = [p_{i1}, p_{i2}, ..., p_{ic}]$.

The node-to-class relational consistency between $\mathbf{p}_i^{local}$ and $\mathbf{p}_i^{global}$ is kept by minimizing the Kullback-Leibler divergence [38] between them as follows:

$$
\begin{aligned}
L_{node2class} &= \sum_{i=1}^{m-l} D_{KL}(\mathbf{p}_i^{local} \parallel \mathbf{p}_i^{global}) \\
&= \sum_{i=1}^{m-l} g(\mathbf{p}_i^{local}, \mathbf{p}_i^{global}) - g(\mathbf{p}_i^{local}),
\end{aligned}
$$

$$
\begin{aligned}
g(\mathbf{p}_i^{local}, \mathbf{p}_i^{global}) &= -\sum_{j=1}^{c} softmax(\mathbf{p}_i^{local}) \\
&\times \ln(softmax(\mathbf{p}_i^{global})),
\end{aligned} \quad (9)
$$

where $g(\mathbf{p}_i^{local})$ is omitted, $softmax(\cdot)$ is the softmax function, and $g(\mathbf{p}_i^{local}, \mathbf{p}_i^{global})$ is implemented by the cross-entropy function according to [39]. In the proposed IMCN model, $L_{node2class}$ is regarded as a middle-grained consistency constraint.

3.4 Feature fusion and node classification

Node representations with the individuality and commonality of nodes are generated by the designed encoders under the defined multi-granularity consistency constraints. That important information is integrated and complements each other to obtain the final node representations as follows:

$$
\mathbf{H} = \lambda \mathbf{H}^{local} + (1 - \lambda)\mathbf{H}^{global}, \quad (10)
$$

where λ is a trade-off hyper-parameter between the individuality and commonality of nodes, and $\lambda \in (0, 1)$.

Then, the embedding vectors of l labeled nodes can be noted as $\mathbf{H}' \in \mathbb{R}^{l \times c}$ from $\mathbf{H}$, and the cross-entropy classification loss is calculated to penalize the differences between the predicted labels $\hat{\mathbf{Y}} = softmax(\mathbf{H}')$ and the ground truth $\mathbf{Y} \in \mathbb{R}^{l \times c}$ of the labeled nodes as follows:

$$
L_{cross} = -\sum_{i=1}^{l} \sum_{j=1}^{c} \mathbf{Y}_{ij} \ln \hat{\mathbf{Y}}_{ij}. \quad (11)
$$

Finally, the proposed semi-supervised classification model IMCN is trained with the overall loss function expressed as follows:

$$
L = L_{cross} + \alpha L_{node} + \beta L_{class} + \gamma L_{node2class}, \quad (12)
$$

where α, β, and γ are three adjustable hyper-parameters to measure the importance of multi-granularity consistencies respectively. The training process of IMCN is sketched in Algorithm 1.

Algorithm 1 Individuality-enhanced and Multi-granularity Consistency-preserving graph neural Network (IMCN).

Require: Graph with adjacency matrix $\mathbf{A}$, node feature matrix $\mathbf{X}$ and one-hot label matrix $\mathbf{Y}$ of labeled nodes, hyper-parameters $\mu, \tau, \lambda, \alpha, \beta, \gamma$, and the number of iterations $\mathcal{T}$;
Ensure: Predicted labels for unlabeled nodes;
1: **for** $t = 1 : \mathcal{T}$ **do**
2:      Extract individual and common feature of nodes by (1) and (2);
3:      Calculate the node-level consistency loss by (3), (4), and (5);
4:      Generate class prototypes for global and local labeled nodes by average aggregation;
5:      Compute and update class prototypes from two views by (7), respectively;
6:      Calculate the class-level consistency loss with (6);
7:      Compute the node-to-class consistency loss for unlabeled nodes from two views by (8) and (9);
8:      Calculate the final node representations according to (10);
9:      Compute the cross-entropy classification loss for labeled training nodes by (11);
10:      Update parameters in the network by minimizing the overall loss function in (12);
11: **end for**
12: Conduct category prediction for unlabeled nodes based on the trained IMCN model;

---

4 Experimental setup

This section presents the experimental setup from the following three perspectives: (1) benchmark datasets used for training and testing the model; (2) baseline models compared with the proposed model; and (3) parameter settings for the proposed model in the series of experiments.

4.1 Datasets

Six benchmark datasets are used in the experiments for a comprehensive comparison between the proposed method and the state-of-the-art methods, including three undirected citation networks from [40], two co-purchasing networks segmented from the Amazon co-purchasing graph [41], and one co-authorship network [42] from the KDD Cup 2016 challenge.[1] Detailed statistics of these datasets are summarized in Table 2, where the density of each dataset is defined as the ratio between the number of actual edges in the dataset and the edges in its corresponding fully connected graph.

Following the data preprocessing in [19], each dataset is split into training, validation, and test sets as follows: (1) For the first three citation networks, twenty labeled nodes per class are used as the training set, 500 nodes, and 1,000 nodes as the validation and test sets, respectively. (2) For the other three networks, thirty labeled nodes per class are used as the training set, thirty nodes per class as the validation set, and the rest as the test set.

Table 2 Dataset statistics

Datasets	Type	Nodes	Features	Edges	Classes	Label rate	Density
Cora	Citation	2,708	1,433	5,429	7	5.2%	0.148%
CiteSeer	Citation	3,327	3,703	4,732	6	3.6%	0.086%
PubMed	Citation	19,717	500	44,338	3	0.3%	0.023%
Photo	Co-purchasing	7,650	745	119,081	8	3.1%	0.407%
Computer	Co-purchasing	13,752	767	245,861	10	2.2%	0.260%
CS	Co-authorship	18,333	6,805	81,894	15	2.5%	0.049%

4.2 Comparison models

To verify the effectiveness of the proposed model, a comparison is made between the proposed IMCN model and ten other baseline methods, including four basic deep graph models (GCN [11], GAT [24], SGC [25], and H-GCN [12]), and six GCN-based contrastive models (DGI [26], GMI [43], MVGRL [21], GRACE [27], CG^3 [19], and VCHN [20]). The description of the details of these methods is as follows.

1) **GCN** [11] produces node embedding vectors by a recursive average neighborhood aggregation scheme. It is derived from the related work of conducting graph convolutions in the spectral domain [44].

2) **GAT** [24] generates node embedding vectors by modeling the differences between the node and its one-hop neighbors.

3) **SGC** [25] reduces the excess complexity in GCN by removing nonlinearities and collapsing weight matrices between consecutive layers.

4) **DGI** [26] generates node embeddings and graph summary vector for the original input graph and constructs a corrupted graph to obtain negative node embeddings with the same GNN encoder. Then DGI aims at maximizing the mutual information between positive node embeddings and the graph summary vector and minimizing it between negative node embeddings and the graph summary vector.

5) **GMI** [43], different from DGI, focuses on maximizing the mutual information of feature and edge between the input graph and the output graph of the encoder.

6) **MVGRL** [21] uses graph diffusion to generate an additional structural view of a graph, then original-view and diffusion-view graphs are fed to GNNs and shared MLP to learn node representations. The learned features are then fed to a graph pooling layer and a shared MLP to learn graph representations. A discriminator contrasts node representations from one view with graph representation of another view and vice versa and scores the agreement between representations which is used as the training signal.

7) **GRACE** [27] jointly corrupts the input graph at both topology and node attribute levels, such as removing edges and masking node features, to provide diverse contexts for nodes in different views. Then contrastive learning is conducted between node embeddings from two views.

8) **H-GCN** [12] is an improved GCN-based model that expands the receptive field of graph convolutions in GCN.

9) **CG3** [19] employs the H-GCN model and a two-layer GCN module to obtain local and global node embeddings and designs a semi-supervised node-level contrastive loss and a graph-level generative loss to optimize the model learning process.

10) **VCHN** [20] uses a two-layer GCN module and a two-layer GAT module to obtain latent features from spectral and spatial views and designs a strategy to generate confident pseudo-labels for unsupervised nodes.

Note that, in the last experiment of Section 5, IMCN with a two-layer GCN as the local encoder is added to illustrate the effectiveness of the proposed scheme in this paper.

4.3 Experimental settings

The proposed IMCN model was trained using the Adam optimizer with 500 epochs and the following settings: (1) The ReLU function is adopted as the non-linear activation of hidden layers. (2) The output dimension of local and global node representations is fixed to the number of classes. The dimensions of hidden layers, learning rate, weight decay, and dropout ratio are searched in {32, 64, 128}, {0.1, 0.05, 0.01}, {0.01, 0.005, 0.001, 0.0005}, and {0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}, respectively. (3) The hyper-parameters μ, λ, and τ in the proposed IMCN model are searched in {0.1, 0.2, 0.3, 0.4, 0.5}. (4) The hyper-parameters α, β, and γ for the trade-off among three consistencies at different granularity are tuned in {0.1, 0.5, 1, 1.5, 2}. In each experiment, the proposed IMCN model is run 10 random trials and the mean and standard deviation of the best test classification accuracy is reported. The results of the comparison methods are directly excerpted from the original papers. If not, corresponding experiments are conducted to obtain the results. The code and datasets are publicly available at https://github.com/xinya0817/IMCN.

5 Experimental results and analysis

This section presents the experimental results and discusses the performance of IMCN from the following seven aspects: (1) Performance of IMCN with different weights of the node-level consistent constraint, the class-level one, and the one at the node-to-class level; (2) Performance of IMCN with different updating rates for learning class prototypes, different temperatures for calculating the node-to-class distribution, and different weights for the local embedding in the final embedding; (3) Visualization of the original nodes and node embeddings learned by IMCN and its part modules; (4) Ablation study of IMCN with different loss terms; (5) Performance of IMCN on alleviating over-smoothing; (6) Performance of IMCN and comparison methods with scarce labeled training data; and (7) Performance of IMCN and baselines on common benchmark datasets. In all tables of experimental results, the highest record on each dataset is highlighted in bold.

5.1 Performance of IMCN with different weights on the consistent constraints

Experiments are carried out to determine the effectiveness of three local-global consistent constraints in IMCN. To verify the performance of the proposed IMCN model on very limited labeled training nodes, two citation networks (a small one and a relatively big one) are used with label rates equal 0.5% for Cora and 0.03% for PubMed. When different values are set for one hyper-parameter, the other two hyper-parameters are fixed.

The classification accuracies of IMCN with different values for α, β, and γ are shown in Fig. 3, and the following observations can be obtained:

(1) IMCN obtained the best classification result on Cora when $\alpha = 1.5$, $\beta = 1$, and $\gamma = 2$. This is significantly superior to the results when α, β, or γ equals 0.1.

(2) IMCN got the best result on PubMed when $\alpha = 0.5$, $\beta = 1$, and $\gamma = 0.1$, which is obviously much better when these hyper-parameters are set to other values. Therefore, the impacts of three different granularity consistency levels on the model are quantified.

(3) The IMCN model demonstrates satisfactory performance when the values of α and β are approximately 1, regardless of whether it is applied to Cora or PubMed datasets. However, the same level of sensitivity was not observed in relation to the parameter γ. This indicates that IMCN is considerably more responsive to the weight of the node-to-class loss. As a result, it is recommended to set the parameters α and β to 1, while the parameter γ may require careful fine-tuning in domain-specific applications.

5.2 Performance of IMCN with different parameters μ, τ, λ

This part discusses the impacts of different rates μ for updating class prototypes, different temperatures τ for the node-to-class distribution, and different weights λ of the local
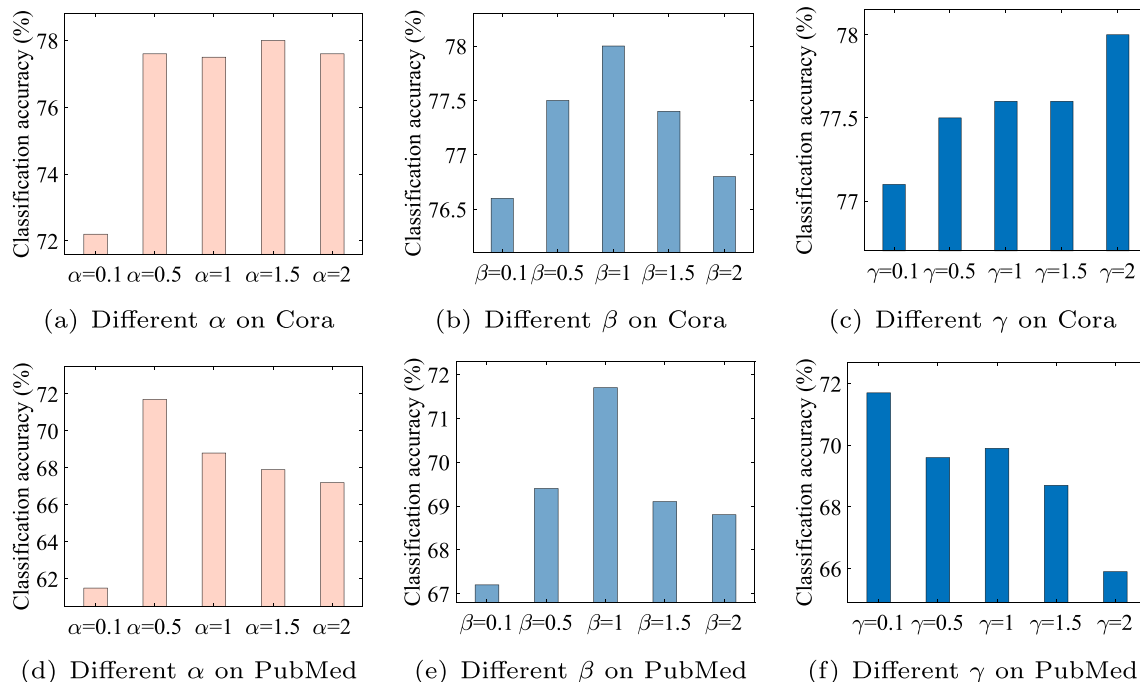

Fig. 3 Classification accuracies of the proposed IMCN model with different α, β, and γ

embedding in the final embedding on the performance of the proposed IMCN model. Classification experiments are conducted on Cora and PubMed with 0.5% and 0.03% labeled nodes, respectively. From the results shown in Fig. 4, the following observations can be obtained:

(1) IMCN performs best when μ is set to 0.3 on Cora and 0.4 on PubMed. Smaller or bigger values for μ cannot ensure IMCN gets the ideal performance, which implies that appropriate updating speed for class prototypes is important for learning stable and expressive node representations. On the one hand, when μ is very small, node embeddings are unable to obtain new useful information in a timely manner. On the other hand, when μ is very large, however, the important information learned previously cannot be retained.

(2) In general, a low value for the temperature hyper-parameter τ ensures that IMCN achieves the best performance, as seen on Cora. This is because that small temperature hyper-parameter can ensure a high concentration of node-to-class distribution. However, the superior result on PubMed obtained when $\tau = 0.3$ in IMCN was used may be attributed to PubMed's relatively large scale and sparse structure.

(3) The best classification accuracy was got when $\lambda = 0.3$ for IMCN both on Cora and PubMed. A much smaller or bigger percentage of local information in the final node representations cannot obtain ideal results. This is because the hierarchical GCN module takes the feature information of the node into learning, but some are damaged by the propagation and aggregation operations of GCN layers.

(4) IMCN demonstrates strong performance when the values of μ and λ are set to 0.4 and 0.3, respectively, whether applied to Cora or PubMed datasets. However, the same level of performance consistency was not observed in relation to the parameter τ. This indicates that IMCN is highly sensitive to the temperature parameter in the node-to-class loss. Therefore, it is recommended to set the parameters μ and λ to 0.4 and 0.3, respectively, while the parameter τ may require fine-tuning in practical tasks.

5.3 Visualization of node embeddings learned by different models

The t-SNE algorithm [45] is used to visualize the original nodes of Cora [40] with a label rate of 0.5% and their embedding representations learned by a two-layer MLP (only the features of node itself are used), the representative model H-GCN [12] (feature information propagated from multi-hop neighbors is used), and the proposed IMCN model (which integrate feature information of node itself and from its neighbors). All original and embedded nodes are projected into a two-dimensional space for visualization and shown in Fig. 5.

From the results, the following observations can be obtained. After the embedding process of a simple two-layer MLP model, nodes from different classes are still mixed and cannot be clearly distinguished. H-GCN can group most embedded nodes into their classes correctly, however, many nodes from different classes in the central area of Fig. 5(c)
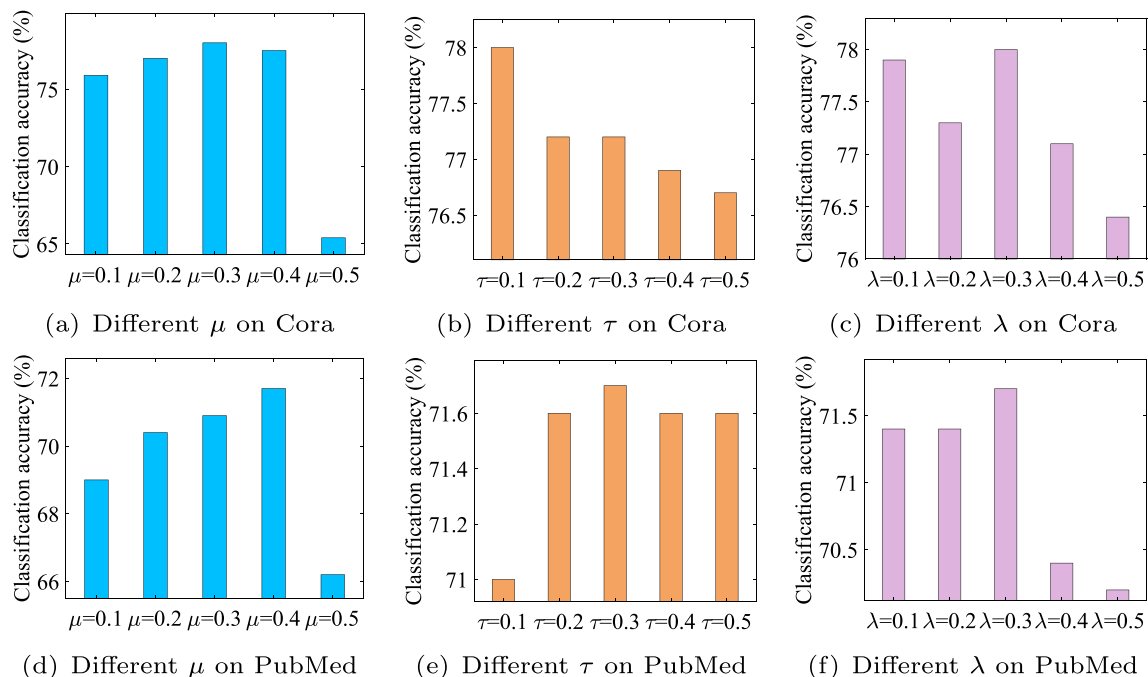

Fig. 4 Classification accuracies of the proposed IMCN model with different μ, τ and λ
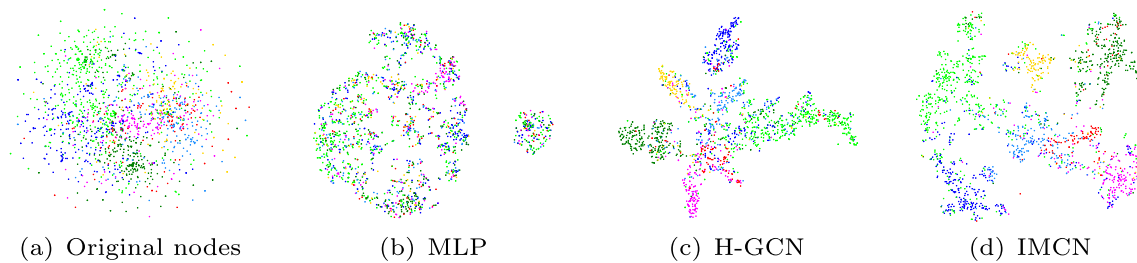
Fig. 5 Two-dimension visualization of original nodes and node embeddings obtained by MLP, H-GCN, and the proposed IMCN model on Cora

are very close, which can easily lead to misclassification. Compared with the above two methods, the proposed IMCN model can push nodes from different classes away while increasing the distance between these classes, ensuring low classification errors.

5.4 Ablation study of IMCN

On three citation networks, ablation experiments are carried out to demonstrate the effectiveness of various local-global consistent constraints in IMCN. The label rates of Cora, Cite-Seer, and PubMed are 0.5%, 0.5%, and 0.03%, respectively. Experimental results are listed in Table 3.

From the results, it can be seen that the designed multi-granularity constraints (feature embedding agreement, semantic class alignment, and the identity of node-to-class relational distribution) make a significant improvement. The consistency of the local and global perspectives at multiple levels can reveal their complementary features between the individuality and commonality of nodes. By combining these constraints, IMCN can make full use of both limited labeled nodes and abundant unlabeled nodes and integrate useful information from the two views.

5.5 Performance of IMCN on alleviating over-smoothing

Through a series of experiments, it was observed that the nine-layer and twelve-layer GCNs cause all node representations to be similar and indistinguishable on Cora and CiteSeer, respectively, as shown in Fig. 6(a) and (c). Then, the

proposed IMCN models with corresponding GCNs as global encoders obtain low-dimensional node representations on these two datasets respectively, as shown in Fig. 6(b) and (d). From these figures, it can be seen that the proposed IMCN method obviously alleviates the over-smoothing problem resulting from multiple convolution operations.

5.6 Performance of IMCN and comparison methods on datasets with scarce labeled nodes

In this part, some experiments are conducted to verify the effectiveness of the proposed IMCN method in learning expressive node embeddings with only a few nodes labeled in the training process. In the experiments, three benchmark graph datasets (Cora, CiteSeer, and PubMed) with different label rates are used: 0.5%, 1%, 3% labeled nodes for Cora and CiteSeer; and 0.03%, 0.05%, 0.01% labeled nodes for PubMed, respectively. The classification accuracies of all methods are listed in Table 4, and the following three observations are obtained.

(1) The proposed IMCN method outperforms most baselines in terms of different label rates on three datasets, especially when there are very few labeled nodes. For example, on CiteSeer with 0.5% labeled nodes, the classification accuracy of IMCN is significantly higher than that of other methods, which is 6.5% higher than the method ranked second. This mainly attributes that IMCN can capture the abundant individuality and commonality of nodes with the consideration of the complex relations among nodes.
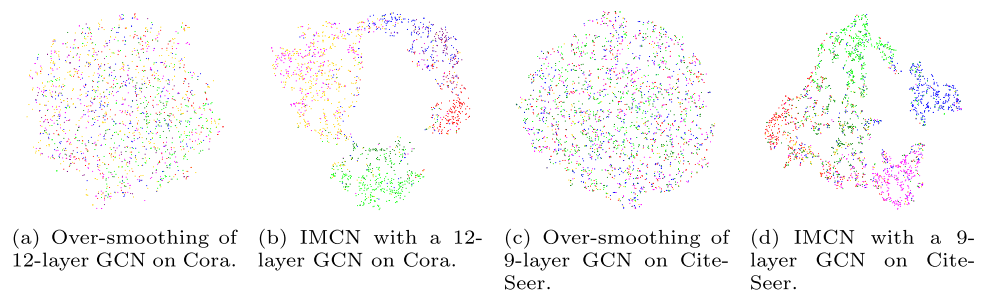
(2) On Cora with a label rate of 1% and PubMed with a label rate of 0.1%, IMCN's performance is not superior to VCHN, which is ranked first, but it is clearly better than the method ranked third. Concretely, the classification accuracy of the proposed IMCN method is 2.6% and 0.1% lower than VCHN, but is 3.7% and 0.4% higher than H-GCN on these two datasets, respectively.

(3) The performance of the proposed IMCN model is quite good when density is high, especially on Cora and CiteSeer. This is obviously in contrast with that on PubMed which is much sparser than the first two datasets according to their density information in Table 2.

Table 3 Ablation study of the proposed IMCN method with different loss terms (%)

L_{cross}	L_{node}	L_{class}	$L_{node2class}$	Cora	CiteSeer	PubMed
✓				68.7	55.4	69.8
✓	✓			70.0	68.4	70.8
✓	✓	✓		76.9	70.0	71.1
✓	✓	✓	✓	**78.0**	**70.8**	**71.7**

Fig. 6 Two-dimension visualization of node embeddings obtained by multi-layer GCNs and the proposed IMCN models with the corresponding multi-layer GCNs on Cora and CiteSeer

(a) Over-smoothing of 12-layer GCN on Cora.

(b) IMCN with a 12-layer GCN on Cora.

(c) Over-smoothing of 9-layer GCN on Cite-Seer.

(d) IMCN with a 9-layer GCN on Cite-Seer.

5.7 Performance of IMCN and baselines on common benchmark datasets

In this section, classification experiments are conducted on six various networks with common dissociation to assess the performance of the proposed IMCN method and compare it with the baseline methods. Note that IMCN1 and IMCN2 are the methods proposed in this paper with the H-GCN model and a two-layer GCN as the global encoder, respectively. In addition, there are two designed comparison methods corresponding to IMCN1 and IMCN2 without multi-granularity consistency constraints: IEN1 takes the H-GCN model and a two-layer MLP as encoders; and IEN2 uses a two-layer GCN and a two-layer MLP as encoders.

First, the classification accuracies of all methods are shown in Table 5, where the results ranking in the first two are marked in bold. The following three conclusions can be drawn:

(1) The performance of IMCN1 and IMCN2 is obviously better than the first three traditional models, GCN, GAT, and SGC, which are based on a single view. This is thanks to two different views of the graph combined in IMCN to capture both shared and complementary information from them.

(2) The proposed methods IMCN1 and IMCN2 outperform the contrastive learning-based comparison methods on all experimental datasets. Especially, IMCN1 and IMCN2

obtain the best classification accuracies on Photo which are about 2.9% and 3.3% higher than that of CG^3 which ranked at third and proposed in recent years. This mainly owes to the individuality of nodes enhanced by the designed simple local encoder and the multi-granularity relations among nodes and classes maintained by the designed consistency constraints.

(3) IMCN1 and IMCN2 are obviously better than the corresponding IEN1 and IEN2 on most datasets. For example, on the citation network CiteSeer, the node classification accuracies are promoted by 3.7% and 5.7% with the designed multi-granularity consistency constraints of the proposed method.

Then, the widely used statistical Nemenyi test [46] is employed to conduct a comprehensive analysis of the significant difference among the proposed IMCN methods and 12 comparison methods on six datasets with the classification accuracies in Table 5. The average ranks of all the methods with the critical distance (CD) are plotted as shown in Fig. 7. The following observations are obtained. The classification accuracies of the proposed IMCN1, IMCN2, IEN1, IEN2, CG^3, H-GCN, and MVGRL are statistically better than those of other seven comparison methods. There is no consistent evidence to indicate the statistical differences among IMCN1, IMCN2, IEN1, IEN2, CG^3, H-GCN, and MVGRL on the metric of classification accuracy.

Table 4 Classification accuracies of the proposed IMCN method and comparison methods on Cora, CiteSeer, and PubMed with very limited labeled nodes (%)

Dataset	Cora			CiteSeer			PubMed		
Label rate	0.5%	1%	3%	0.5%	1%	3%	0.03%	0.05%	0.1%
GCN [11]	42.6	56.9	74.9	33.4	46.5	66.9	61.8	68.8	71.9
GAT [24]	56.4	71.7	78.5	45.7	64.7	69.3	65.7	69.9	72.4
SGC [25]	43.7	64.3	71.0	43.2	50.7	60.9	62.5	69.4	69.9
DGI [26]	67.5	72.4	78.9	60.7	66.9	69.8	60.2	68.4	70.7
GMI [43]	67.1	71.0	78.8	56.2	63.5	68.0	60.1	62.4	71.4
MVGRL [21]	61.6	65.2	79.0	61.7	66.6	70.3	63.3	69.4	72.2
GRACE [27]	60.4	70.2	75.8	55.4	59.3	67.8	64.4	67.5	72.3
H-GCN [12]	70.9	75.0	82.5	56.3	60.3	69.6	68.7	70.3	74.7
CG^3 [19]	69.3	74.1	79.9	62.7	70.6	71.3	68.3	70.1	73.2
VCHN [20]	73.9	**81.3**	82.0	64.3	67.9	69.1	69.2	71.8	**75.2**
IMCN (our)	**78.0**	78.7	**83.3**	**70.8**	**72.9**	**73.8**	**71.7**	**72.4**	75.1

Table 5 Classification accuracies of the proposed IMCN method and comparison methods on six benchmark datasets (%)

Methods	Cora	CiteSeer	PubMed	Photo	Computer	CS
GCN [11]	81.5±0.0	70.3±0.0	79.0±0.0	87.3±1.0	76.3±0.5	91.8±0.1
GAT [24]	83.0±0.7	72.5±0.7	79.0±0.3	86.2±1.5	79.3±1.1	90.5±0.7
SGC [25]	81.0±0.0	71.9±0.1	78.9±0.0	86.4±0.0	74.4±0.1	91.0±0.0
DGI [26]	81.7±0.6	71.5±0.7	77.3±0.6	83.1±0.5	75.9±0.6	90.0±0.3
GMI [43]	82.7±0.2	73.0±0.3	80.1±0.2	85.1±0.1	76.8±0.1	91.0±0.0
MVGRL [21]	82.9±0.7	72.6±0.7	79.4±0.3	87.3±0.3	79.0±0.6	91.3±0.1
GRACE [27]	80.0±0.4	71.7±0.6	79.5±1.1	81.8±1.0	71.8±0.4	90.1±0.8
H-GCN [12]	82.9±0.3	71.1±0.5	79.9±0.4	92.0±0.8	80.4±0.8	92.5±0.1
CG^3 [19]	83.4±0.7	73.6±0.8	80.2±0.8	89.4±0.5	79.9±0.6	92.3±0.2
VCHN [20]	81.6±0.5	71.5±0.6	78.7±0.4	89.3±1.2	82.1±0.3	88.4±0.3
IEN1	83.2±0.4	72.8±0.3	79.4±0.4	93.0±0.3	83.1±0.3	93.2±0.2
IMCN1 (our)	**84.6±0.4**	**76.5±0.3**	**81.2±0.8**	**93.1±0.5**	83.6±0.4	**93.3±0.2**
IEN2	82.0±0.2	71.2±0.9	79.3±0.5	92.8±0.2	**84.1±0.5**	93.2±0.2
IMCN2 (our)	**85.1±0.2**	**76.9±0.2**	**82.2±1.2**	**93.6±0.2**	**84.9±0.4**	**93.7±0.2**

Fig. 7 Average ranks of all methods with the critical distance (CD) for classification accuracy according to the Nemenyi test [46]
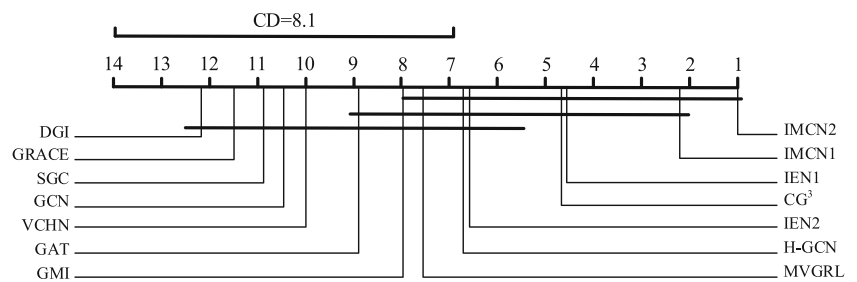
Table 6 Macro-F_1 results of the proposed IMCN methods and comparison methods on six benchmark datasets (%)

Methods	Cora	CiteSeer	PubMed	Photo	Computer	CS
GCN [11]	76.5±0.5	60.8±1.1	77.1±0.5	84.4±1.3	69.1±6.4	80.2±0.9
H-GCN [12]	81.9±0.3	67.8±0.6	79.2±0.4	90.2±1.0	76.8±1.7	90.7±0.1
CG^3 [19]	81.4±0.5	71.3±0.4	80.2±0.2	90.3±1.0	77.0±3.3	90.7±0.1
VCHN [20]	80.7±0.7	67.6±0.6	77.9±0.3	87.3±0.9	82.3±0.3	85.4±0.4
IEN1	82.0±0.5	68.9±0.4	78.9±0.4	91.4±0.4	84.4±0.4	91.5±0.3
IMCN1 (our)	83.6±0.4	71.0±0.4	80.0±0.7	91.2±0.5	81.9±0.8	91.5±0.2
IEN2	80.9±0.3	67.3±0.7	78.9±0.5	91.1±0.2	84.1±0.4	91.5±0.2
IMCN2 (our)	**84.0±0.3**	**71.8±0.4**	**81.3±0.9**	**91.9±0.3**	**84.5±0.4**	**92.1±0.2**

Table 7 FLOPs, trainable parameters, test time, and running memory of GCN, H-GCN, and the corresponding IMCNs

Metrics	Datasets	H-GCN [12]	IMCN1 (our)	GCN [11]	IMCN2 (our)
FLOPs	Cora	0.70G	1.09G	0.77G	0.04G
	CiteSeer	0.94G	1.47G	0.77G	0.05G
	PubMed	4.58G	8.51G	1.17G	1.21G
Parameters	Cora	0.49M	0.59M	0.09M	0.18M
	CiteSeer	1.07M	0.58M	0.09M	0.47M
	PubMed	0.25M	1.31M	0.03M	0.06M
Test time	Cora	0.04s	0.04s	0.01s	0.02s
	CiteSeer	0.05s	0.07s	0.01s	0.04s
	PubMed	0.26s	0.91s	0.40s	0.95s
Memory	Cora	1.52G	1.10G	0.11G	0.40G
	CiteSeer	1.66G	1.21G	0.11G	0.50G
	PubMed	2.97G	3.98G	0.41G	1.63G

According to the ranks of the proposed methods and 12 comparison methods, macro-F_1 results of the first six methods (including IMCN1, IMCN2, IEN1, IEN2, CG^3 and H-GCN), the latest method VCHN, and the baseline GCN are listed and compared in Table 6. The best results are marked in bold. It can be seen that, under the macro-F_1 metric, IMCNs' performance is much better than that of their corresponding IENs, GCN, H-GCN, and VCHN in most cases, especially obvious on Cora. This is mainly attributed to the specific designed individuality-enhanced module and three consistency constraints at different levels.

Finally, Table 7 shows the FLOPs, trainable parameters, test time, and running memory of the proposed methods IMCN1 and IMCN2 and the corresponding baseline methods (H-GCN and GCN). It can be seen that the space and time complexity of the proposed methods is slightly higher than the corresponding baseline methods while maintaining competitive performance. This is because of the added two-layer MLP to enhance individuality and the three consistency constraints between local and global encoders.

6 Conclusions and future work

In this paper, we proposed a graph neural network called Individuality-enhanced and Multi-granularity Consistency-preserving Network (IMCN). IMCN aims to take advantage of the limited, yet valuable, supervised information available in labeled data and effectively enhance the classification capability. On the one hand, a simple MLP module was combined with the original GCN-based model to enhance individual information in learning node representations. On the other hand, the complex relations among nodes and classes were taken full advantage of by the designed three consistency constraints for optimizing the encoding processes of two encoders. Extensive experiments were conducted on various public benchmark datasets, and the results demonstrated the effectiveness of the proposed IMCN method in solving node classification tasks with extremely limited labeled nodes.

The proposed IMCN model has strict requirements on the input graph, which assumes that the entire structure of this graph is available to capture the common-feature information of nodes. Moreover, IMCN has a considerable number of hyperparameters that require tuning and can be inefficient when dealing with very large networks. In the future, our focus will be on developing scalable and efficient deep semi-supervised node classification methods specifically designed for large-scale graph datasets. We also aim to explore automatic parameter tuning techniques using optimization methods, such as [47].

Declarations

References

1. Wang K, An J, Zhou M, Shi Z, Shi X, Kang Q (2023) Minority-weighted graph neural network for imbalanced node classification in social networks of internet of people. IEEE Internet Things J 10(1):330–340

2. Yu H, Shen Z, Du P (2022) NPI-RGCNAE: fast predicting ncRNA-protein interactions using the relational graph convolutional network autoencoder. IEEE J Biomed Health Inform 26(4):1861–1871

3. Zheng Y, Gao C, He X, Jin D, Li Y (2023) Incorporating price into recommendation with graph convolutional networks. IEEE Trans Knowl Data Eng 35(2):1609–1623

4. Liu J, Xia F, Feng X, Ren J, Liu H (2022) Deep graph learning for anomalous citation detection. IEEE Trans Neural Netw Learn Syst 33(6):2543–2557

5. Lin G, Kang X, Liao K, Zhao F, Chen Y (2021) Deep graph learning for semi-supervised classification. Pattern Recognit 118:108039

6. Yang F, Zhang H, Tao S (2022) Simplified multilayer graph convolutional networks with dropout. Applied Intelligence 52(5):4776–4791

7. Wang J, Liang J, Cui J, Liang J (2021) Semi-supervised learning with mixed-order graph convolutional networks. Information Sciences 573:171–181

8. Li K, Ye W (2022) Semi-supervised node classification via graph learning convolutional neural network. Applied Intelligence 52(11):12724–12736

9. Kazi A, Cosmo L, Ahmadi SA, Navab N, Bronstein MM (2023) Differentiable graph module (DGM) for graph convolutional networks. IEEE Trans Pattern Anal Mach Intell 45(2):1606–1617

10. Zhou H, Gong M, Wang S, Gao Y, Zhao Z (2023) SMGCL: semisupervised multi-view graph contrastive learning. Knowledge-Based Systems 260:110120

11. Kipf TN, Welling M (2017) Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations

12. Hu F, Zhu Y, Wu S, Wang L, Tan T (2019) Hierarchical graph convolutional networks for semi-supervised node classification. In: International Joint Conference on Artificial Intelligence, pp 4532–4539

13. Li Q, Han Z, Wu X (2018) Deeper insights into graph convolutional networks for semi-supervised learning. In: AAAI Conference on Artificial Intelligence, pp 3538–3545

14. Rong Y, Huang W, Xu T, Huang J (2019) Dropedge: towards deep graph convolutional networks on node classification. In: International Conference on Learning Representations

15. Chen M, Wei Z, Huang Z, Ding B, Li Y (2020) Simple and deep graph convolutional networks. International Conference on Machine Learning 119:1725–1735

16. He K, Fan H, Wu Y, Xie S, Girshick RB (2020) Momentum contrast for unsupervised visual representation learning. In: IEEE Conference on Computer Vision and Pattern Recognition, pp 9726–9735

17. Chen X, Yao L, Zhou T, Dong J, Zhang Y (2021) Momentum contrastive learning for few-shot COVID-19 diagnosis from chest CT images. Pattern Recognition 113:107826

18. Xu H, Xiong H, Qi G (2022) K-shot contrastive learning of visual features with multiple instance augmentations. IEEE Trans Pattern Anal Mach Intell 44(11):8694–8700

19. Wan S, Pan S, Yang J, Gong C (2021) Contrastive and generative graph convolutional networks for graph-based semi-supervised learning. In: AAAI Conference on Artificial Intelligence, pp 10049–10057

20. Liao Z, Zhang X, Su W, Zhan K (2022) View-consistent heterogeneous network on graphs with few labeled nodes. IEEE Transactions on Cybernetics 1–10

21. Hassani K, Ahmadi AHK (2020) Contrastive multi-view representation learning on graphs. In: International Conference on Machine Learning, pp 4116–4126

22. Song Y, Gu Y, Li X, Li C, Yu G (2022) CSGNN: improving graph neural networks with contrastive semi-supervised learning. Int Conf Adv Comput Appl 13245:731–738

23. Hamilton WL, Ying Z, Leskovec J (2017) Inductive representation learning on large graphs. In: Annual Conference on Neural Information Processing Systems, pp 1024–1034

24. Velickovic P, Cucurull G, Casanova A, Romero A, Lió P, Bengio Y (2018) Graph attention networks. In: International Conference on Learning Representations

25. Wu F Jr, AHS, Zhang T, Fifty C, Yu T, Weinberger KQ, (2019) Simplifying graph convolutional networks. International Conference on Machine Learning 97:6861–6871

26. Velickovic P, Fedus W, Hamilton WL, Lió P, Bengio Y, Hjelm RD (2019) Deep graph infomax. In: International Conference on Learning Representations

27. Zhu Y, Xu Y, Yu F, Liu Q, Wu S, Wang L (2020) Deep graph contrastive representation learning. CoRR abs/2006.04131

28. Liu Y, Li Z, Pan S, Gong C, Zhou C, Karypis G (2022) Anomaly detection on attributed networks via contrastive self-supervised learning. IEEE Trans Neural Netw Learn Syst 33(6):2378–2392

29. Liu Y, Wang K, Liu L, Lan H, Lin L (2022) TCGL: temporal contrastive graph for self-supervised video representation learning. IEEE Trans Image Process 31:1978–1993

30. Song X, Jin Z (2022) Robust label rectifying with consistent contrastivelearning for domain adaptive person re-identification. IEEE Trans Multimedia 24:3229–3239

31. Lin Y, Gou Y, Liu X, Bai J, Lv J, Peng X (2023) Dual contrastive prediction for incomplete multi-view representation learning. IEEE Trans Pattern Anal Mach Intell 45(4):4447–4461

32. Tian R, Shi H (2023) Momentum memory contrastive learning for transferbased few-shot classification. Applied Intelligence 53(1):864–878

33. You Y, Chen T, Sui Y, Chen T, Wang Z, Shen Y (2020) Graph contrastive learning with augmentations. In: Annual Conference on Neural Information Processing Systems

34. Grover A, Leskovec J (2016) Node2vec: scalable feature learning for networks. In: ACM International Conference on Knowledge Discovery and Data Mining, pp 855–864

35. Nair V, Hinton GE (2010) Rectified linear units improve restricted Boltzmann machines. In: International Conference on Machine Learning, pp 807–814

36. Yang X, Deng C, Dang Z, Wei K, Yan J (2021) SelfSAGCN: Self-supervised semantic alignment for graph convolution network. In: IEEE Conference on Computer Vision and Pattern Recognition, pp 16775–16784

37. Rippel O, Paluri M, Dollár P, Bourdev LD (2016) Metric learning with adaptive density discrimination. In: International Conference on Learning Representations

38. Kullback S, Leibler RA (1951) On information and sufficiency. Ann Math Stat 22(1):79–86

39. Zheng M, You S, Wang F, Qian C, Zhang C, Wang X, Xu C (2021) ReSSL: relational self-supervised learning with weak augmentation. In: Annual Conference on Neural Information Processing Systems, pp 2543–2555

40. Sen P, Namata G, Bilgic M, Getoor L, Gallagher B, Eliassi-Rad T (2008) Collective classification in network data. AI Magazine 29(3):93–106

41. McAuley JJ, Targett C, Shi Q, van den Hengel A (2015) Image-based recommendations on styles and substitutes. In: International Conference on Research and Development in Information Retrieval, pp 43–52

42. Shchur O, Mumme M, Bojchevski A, Gunnemann S (2018) Pitfalls of graph neural network evaluation. CoRR abs/1811.05868
43. Peng Z, Huang W, Luo M, Zheng Q, Rong Y, Xu T, Huang J (2020) Graph representation learning via graphical mutual information maximization. In: International World Wide Web Conferences, pp 259–270
44. Defferrard M, Bresson X, Vandergheynst P (2016) Convolutional neural networks on graphs with fast localized spectral filtering. In: Annual Conference on Neural Information Processing Systems, pp 3837–3845
45. Van der Maaten L, Hinton G (2008) Visualizing data using t-SNE. J Mach Learn Res 9(11):2579–2605
46. Demsar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30
47. Chauhan S, Singh M, Aggarwal AK (2023) Designing of optimal digital IIR filter in the multi-objective framework using an evolutionary algorithm. Eng Appl Artif Intell 119:105803

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Weiren Yu received the Ph.D. degree in computer science from the University of New South Wales, Sydney NSW, Australia. After that, he spent two years as a Postdoctoral Researcher at Imperial College, London, U.K., and then a Lecturer of Computer Science with Aston University, Birmingham, U.K. He is currently an Assistant Professor of Computer Science with the University of Warwick, Coventry, U.K., and an Honorary Visiting Fellow with Imperial College London. His current research interests include graph data management, data mining, and Internet of Things.

Xinxin Liu received a master's degree in computer applied technology from Minnan Normal University, Zhangzhou, China, in 2020. She is currently pursuing the Ph.D. degree in computer science and technology at Nanjing University of Science and Technology, Nanjing, China. Her research interests include graph data mining and machine learning.