

Contrastive Graph Convolutional Networks with Generative Adjacency Matrix

Luying Zhong, Jinbin Yang, Zhaoliang Chen and Shiping Wang

Abstract—Semi-supervised node classification with Graph Convolutional Network (GCN) is an attractive topic in social media analysis and applications. Recent studies show that GCN-based classification methods can facilitate the accuracy increase of learning algorithms. However, most of the existing methods do not conduct adequate explorations of the complementary information within the topology structure. Besides, they also suffer from the insufficient excavation of useful information among nodes and the scarcity of labeled samples, resulting in undesired classification performance. To cope with these issues, this paper proposes a contrastive GCN-based framework to jointly leverage the topology graph and the self-adaptive topology graph with feature information in semi-supervised information. In order to extract more valid potential information in the topology graph and increase the flexibility of the framework, we learn an adjacency matrix supervised by a flexible loss that exploits node embeddings to reinforce the topological representation capability of the adjacency matrix. To maximize the homogeneity of these two distinct graphs, we design an improved semi-supervised contrastive loss. In order to enrich scarce label information, we propose a self-supervised mechanism to generate reliable pseudo labels from abundant unlabeled data, which further refines the learnable adjacency matrix. With these modules, both unlabeled and labeled samples jointly furnish the supervision signals, thereby improving the accuracy of the proposed model. Extensive experimental results on real-world datasets demonstrate the effectiveness and superiority of the proposed algorithm against state-of-the-arts.

Index Terms—Graph convolutional networks, semi-supervised classification, generative adjacency matrix, contrastive learning, self-supervised learning.

I. INTRODUCTION

WITH powerful capabilities of feature representation, graphs have been leveraged to depict a wide range of objects in various data processing fields, such as social network analysis [1], [2], [3], physical systems [4], [5] and computer vision [6], [7], [8]. As an important technique to explore graphs, GCN [9] has been extensively used in considerable practical applications, such as link prediction [10], [11], knowledge graph [12], [13], [14], and node clustering [15], [16], [17], [18] etc. As the extension of convolutional neural networks from Euclidean to non-Euclidean domain, GCN

This work is in part supported by the National Natural Science Foundation of China under Grants U21A20472 and 62276065, and the National Key Research and Development Plan of China under Grant 2021YFB3600503.

Luying Zhong, Jinbin Yang, Zhaoliang Chen and Shiping Wang are with the College of Computer and Data Science, Fuzhou University, Fuzhou 350116, China and also with the Fujian Provincial Key Laboratory of Network Computing and Intelligent Information Processing, Fuzhou University, Fuzhou 350116, China (email: luyingzhongfzu@163.com, yangjinbinfzu@163.com, chenzl23@outlook.com, shipingwangphd@163.com).

(Corresponding author: Shiping Wang.)

generally calculates the embedding of nodes with graph convolutional layers, thereby delivering valid information among neighbors. The typical GCN and its variants [3], [19], [20] employed spectral theory to apply traditional CNN to the non-Euclidean structure. Extensive experimental results indicate that GCNs remarkably surpass traditional graph embedding-based methods. In terms of node classification [21], [22], GCNs reconcile both topology structures and node features to improve the performance of semi-supervised classification with limited supervision signals. To enrich graph structures, some self-supervised methods to study underlying information from unlabeled data have been proposed. A typical technique is contrastive learning [23], [24] which is able to generate effective data representations by exploring the information between similarities and dissimilarities among a set of both unlabeled examples and labeled examples.

Although GCN has gained great success in semi-supervised learning, it still suffers from limited supervision information, which poses a great difficulty for network training and probably degrades the performance of GCNs. To overcome the limited signals, prior works [25], [26] developed a self-training GCN-based model by assigning pseudo labels to top k confident unlabeled nodes. However, the key to generating pseudo labels is the high confidence, and it is inadequate for these works to only select top k confident unlabeled nodes as the condition of generating pseudo data in downstream tasks. Besides, the node relationships in the initial training iterations are mostly unreliable due to the instability of the model, which causes undesired performance. In order to generate reliable pseudo information, more restrictions are imposed to improve the trustworthiness of selected nodes.

Besides, prior studies suggested that it is difficult for feature and topology structures to be optimally integrated into a complex graph. For example, a GCN-based framework named AM-GCN [27] exploited feature similarity between nodes to extract complementary feature information when performing node propagation on a topology network. Wan et al. [28] proposed a contrastive GCN-based method by employing a semi-supervised contrastive loss to extract the potential relationship between graph and data features. However, it is not enough for these works to obtain only potential feature information without concentrating on implicit information from topology spaces, leading to the insufficient co-optimization of feature and graph fusion learning. Moreover, the original data containing sampling noise and useless information may lead to performance degradation. To effectively fuse feature and topology spaces, the potential information without the interference of noise needs to be extracted from feature and

graph structures to enrich deep correlation information.

In order to address these issues, this paper proposes a parallel flexible GCN-based framework named Contrastive Graph Convolutional Networks with generative adjacency matrix (CGCN) to jointly leverage the topology and feature graph refined via node features for semi-supervised learning. Especially, to optimally integrate node features and topology structure via minimizing the interference of sampling noise, CGCN considers to generate a latent adjacency matrix for exploring both implicit topology and feature information. To ensure the reliability of the model, more constraints are imposed to improve the trustworthiness of selected nodes. Therefore, as demonstrated in Fig. 1, CGCN includes two primary modules. One is the generation of a self-adaptive adjacency matrix network with a flexible loss (SA-network) to optimally integrate topology and feature spaces without the interference of noise and useless information. The other is the joint GCN network containing a pseudo-labeling mechanism and semi-supervised loss (PL-network) to generate reliable pseudo signals and increase the robustness of the model.

The whole training procedure can be divided into three fundamental stages. Inspired by the alternating training of two networks for generative adversarial networks, the first stage employs an autoencoder to extract complementary information from a topology space. To generate useful and reliable complementary topology information, a flexible evaluator is applied to assess the quality of extra information. Then we fuse the complementary feature information constructed by the k NN method from node features. The second stage applies an improved semi-supervised contrastive loss to maximize the consistency of embeddings yielded from two distinct graphs. The final stage uses a self-supervised mechanism to capture the signals of mutual supervision and generates feedback to refine the inputs. Especially, the self-supervised mechanism utilizes the reliable pseudo signals owing to model stabilization, high threshold setting, and alignment mechanism to generate feedback for refining the inputs.

Consequently, CGCN is an adjustable model that explores a learnable adjacency matrix and generates pseudo labels with contrastive learning. The main contributions of this paper are summarized from the following four aspects:

- Propose a contrastive graph convolutional network framework to propagate representations across the topology and feature graphs, where semi-supervised contrastive loss is defined to maximize the consistency of graphs.
- Construct a self-adaptive adjacency matrix via adopting an autoencoder with a flexible loss, which supervises the generative process by fusing node features.
- Design two networks to generate pseudo labels during the training process, on the basis of which a self-supervised strategy for supervision information enrichment is constructed with contrastive learning.
- Substantial experiments on benchmark datasets show that the proposed method outperforms state-of-the-art GCN-based methods in terms of semi-supervised classification.

The rest of this paper is organized as follows. Related works on GCN are reviewed in Section II. We elaborate on the proposed CGCN in Section III, including the detailed

introduction of each component and algorithm analyses. Finally, the effectiveness of the proposed framework is verified via substantial experiments in Section IV, and our work is concluded in Section V.

II. RELATED WORK

In this section, we review several representative works on GCN and unsupervised learning techniques.

A. Graph Convolutional Network

Graph-based semi-supervised learning has been a popular branch for the past two decades. It aims to utilize scarce labeled data to achieve the purpose of classification on massive unlabeled nodes. This is usually realized by the low-dimensional embeddings with Laplacian eigenmaps [29] and Markov random walks [30], etc. Especially, spectral networks [31] defined convolutional operations by decomposing a graph signal $\mathbf{s} \in \mathbb{R}^n$ on Fourier domain and then applying a spectral filter \mathbf{g}_θ to the spectral components, formulated as

$$\mathbf{g}_\theta * \mathbf{s} = \mathbf{U} \mathbf{g}_\theta(\Lambda) \mathbf{U}^T \mathbf{s}, \quad (1)$$

where $*$ denotes the convolutional operation, and the normalized graph Laplacian matrix $\mathbf{L} = \mathbf{I}_N - \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2} = \mathbf{U} \theta(\Lambda) \mathbf{U}^T$, \mathbf{D} represents the degree matrix, \mathbf{A} and Λ are denoted as the adjacency matrix and the diagonal matrix of eigenvalues, respectively. Further, David et al. replaced $\mathbf{g}_\theta(\Lambda)$ with the K -th Chebyshev polynomial $\mathbf{T}(\mathbf{s})$, represented as

$$\mathbf{g}_\theta * \mathbf{s} = \sum_{i=0}^K \theta_i \mathbf{T}_i(\mathbf{L}) \mathbf{s}, \quad (2)$$

where θ_i is the coefficient vector of the i -th order Chebyshev polynomial $\mathbf{T}_i(\mathbf{L})$. For the purpose of saving computational resources, Kipf et al. [9] performed the first-order approximation of truncated Chebyshev polynomial [32], i.e.,

$$\mathbf{Z} = \sigma(\tilde{\mathbf{D}}^{-1/2} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-1/2} \mathbf{X} \Theta), \quad (3)$$

where $\sigma(\cdot)$ denotes an activation function, $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_N$, $\tilde{\mathbf{D}}_{ii} = \sum_j \tilde{\mathbf{A}}_{ij}$, $\mathbf{X} \in \mathbb{R}^{N \times C}$ with C channels, $\Theta \in \mathbb{R}^{C \times F}$ represents a trainable parameter matrix and $\mathbf{Z} \in \mathbb{R}^{N \times F}$ is the embedding after convolution.

Due to the outperformance of GCN, considerable variants of GCN have been explored. For example, Liu et al. [33] integrated GCN with a hidden conditional random field to reserve the skeleton structure information. Xu et al. [34] put forward an answer-centric radial graph convolutional networks to cope with the visual question generation tasks. Lei et al. [35] established the graph receptive fields according to diffusion paths and applied them to build a compact graph convolutional network. A multi-stage GCN-based framework was presented by Sun et al. [25] with the self-supervised learning to improve the generalization performance on the graph with limited supervision information. Bo et al. [19] designed a frequency adaptation GCN framework to perceptively combine the low-frequency and high-frequency signals, and improved the performance of the GCN model. These GCN-based works have significantly promoted the performance of different learning tasks in both Euclidean and non-Euclidean domains.

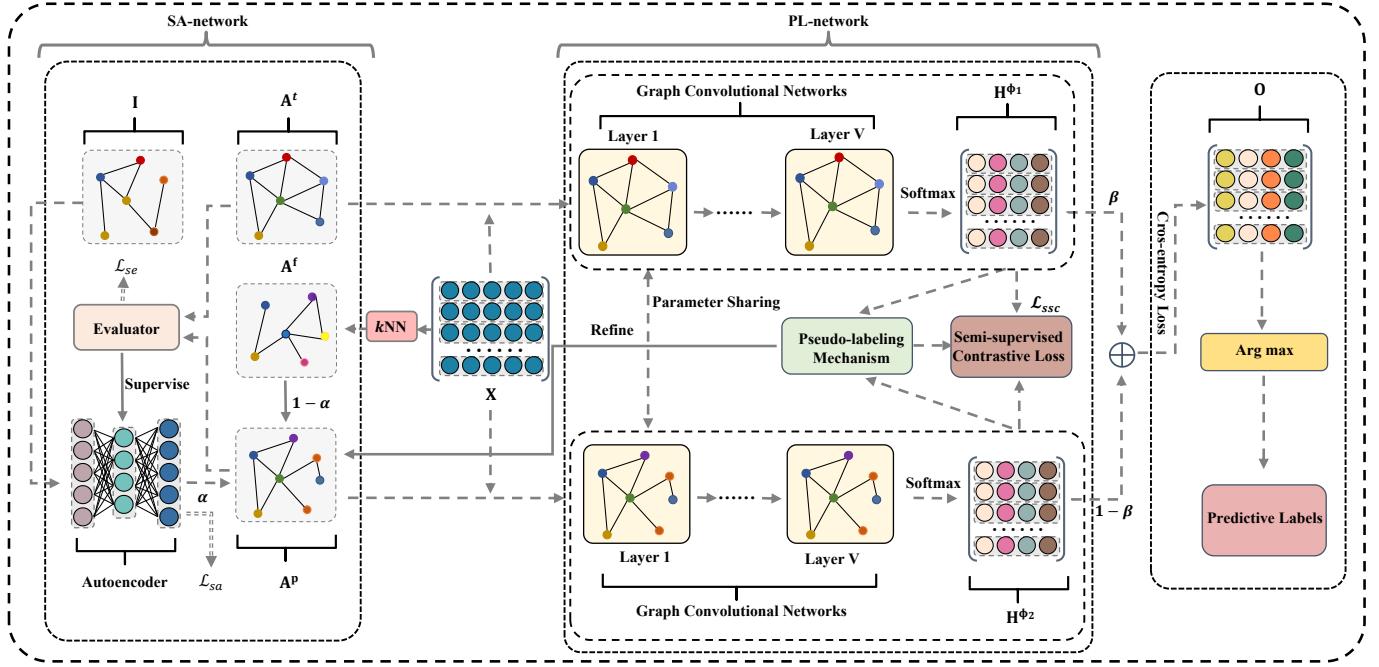


Fig. 1: Architecture of the proposed CGCN, which is a flexible framework of parallel GCNs w.r.t. feature and topology graphs. The networks involve two essential parts: SA-network and PL-network. The training procedure contains three fundamental stages. First, we take the identity matrix as an input of the autoencoder with \mathcal{L}_{sa} being the loss function. A flexible loss supervises the output of the autoencoder, and then fuses \mathbf{A}^f constructed by the kNN method. Second, both \mathbf{A}^p and \mathbf{A}^t are fed to GCNs with a semi-supervised contrastive loss. Finally, a self-supervised mechanism is employed to generate a feedback to refine inputs. To this end, intact representations of both graphs and node features are learned simultaneously to promote the performance of GCN.

B. Unsupervised Learning

Unsupervised learning [36] is a type of machine learning algorithms that seek for patterns from data without known labels, as it can capture rich information contained in the data to guide the representation learning. Compared with supervised learning, unsupervised learning does not rely on any labeled samples and finds patterns by mining the intrinsic features of the data, which is able to identify undetected patterns. Unsupervised learning models like contrastive learning and generative learning can be employed in clustering [37], association [38] and dimensionality reduction [39], etc. In particular, contrastive learning [40] aims to train an encoder to learn the feature representation of samples by comparing the data with positive samples and negative samples in the feature space. He et al. [41] designed an unsupervised momentum method for representation learning, and experimental results demonstrated that the ImageNet classification with MoCo can exceed the performance of supervised learning. Chen et al. [23] proposed a visual representation learning framework, and showed that the effect of this model was close to supervised models. Caron et al. [42] utilized a multi-crop image transformation method and introduced clustering into the model, thus decreasing the computation complexity. Kaveh et al. [43] learned node-level and graph-level representations by contrasting different structures of a graph.

Generative learning is a machine learning method that

explores the high-level semantics similar to the training data to extract valid information. Autoencoder is one of the generative methods which aims to encode the original data, perform dimensionality reduction, and discover patterns between the data. Wang et al. [44] proposed a semi-supervised deep model which used an autoencoder and exploited the first-order proximity to capture non-linear network structures. Wen et al. [45] designed a deep embedding network by encoding each vertex as a low-dimensional vector representation with adversarially regularized autoencoders to enhance the generalization capability. Ke et al. [46] employed two equivalence definitions containing structural equivalence and regular equivalence to extract more discriminative node representations. These generative learning algorithms obtain extra valuable information via exploring deeper information from the original data, thereby enhancing the model performance.

III. THE PROPOSED METHOD

A. Overview and Notations

For the purpose of extracting more complementary information from the topology space and distilling high-confident underlying information, we develop a learnable and contrastive graph convolutional network framework consisting of two primary components: SA-network and PL-network. These modules are divided into three stages of training. First, we use

TABLE I: A summary of primary notations in this paper.

Notations	Explanations	Notations	Explanations
Ψ_l	Labeled set with l samples.	$\mathbf{W}^v, \mathbf{b}^v$	Trainable weight and bias of the v -th layer of autoencoder.
Ψ_u	Unlabeled set with u samples.	$\mathbf{W}_e^v, \mathbf{b}_e^v$	Trainable weight and bias of the v -th layer of evaluator.
Ψ	Set of n samples with $n = l + u$.	\mathbf{A}^s	Pseudo adjacency matrix.
\mathbf{X}	Feature matrix with $n \times d$.	\mathbf{A}^p	Latent adjacency matrix.
\mathbf{Y}	Label matrix with $n \times c$.	$\mathbf{H}^{\Phi_1}, \mathbf{H}^{\Phi_2}$	Embeddings of original and generative graphs.
\mathbf{A}^t	Original adjacency matrix with $n \times n$.	\mathcal{L}_{ssc}	Semi-supervised contrastive loss function.
$\sigma(\cdot)$	Activation function.	\mathcal{L}_{en}	Cross-entropy loss function.
$\mathbf{A}^e, \mathbf{A}^f$	Complementary topology and feature adjacency matrices.	\mathbf{O}	Predictive output of the overall framework.
$\mathcal{L}_{sa}, \mathcal{L}_{se}$	Loss functions of autoencoder and evaluator.	$\mathbf{W}^{(i)}$	Trainable weight matrix during graph convolutions.

SA-network to generate the complementary topology adjacency matrix via node features, then employ the pseudo-labeling mechanism to achieve reliable pseudo information. Finally, a semi-supervised loss is adopted to maximize the consistency of the two views through PL-network. They make up an overall training scheme of the whole network.

Suppose that $\Psi = \{\mathbf{x}_1, \dots, \mathbf{x}_l, \mathbf{x}_{l+1}, \dots, \mathbf{x}_n\}$ is a set with n samples, where l labeled samples form a labeled set Ψ_l and the remaining $u = n - l$ unlabeled samples form an unlabeled set Ψ_u with $l \ll u$. Besides, we denote $\mathbf{X} \in \mathbb{R}^{n \times d}$ as the feature matrix, whose i -th row consists of the feature vector \mathbf{x}_i of the i -th node and d denotes the feature dimension. Label matrix is represented as $\mathbf{Y} \in \mathbb{R}^{n \times c}$ with $\mathbf{Y}_{ij} = 1$ indicating that the i -th node belongs to the j -th class. The symbol c represents the number of classes. We denote the adjacency matrix as $\mathbf{A}^t \in \mathbb{R}^{n \times n}$ with $\mathbf{A}_{ij}^t = 1$ if there is an edge between the i -th node and the j -th node, and $\mathbf{A}_{ij}^t = 0$ otherwise. The feature matrix \mathbf{X} and the adjacency matrix \mathbf{A}^t construct the topology graph $\mathbf{G}_t(\mathbf{X}, \mathbf{A}^t)$.

So as to further clarify the mathematical notation usages in this paper, the explanations of primary notations are listed in Table I.

B. Generative Adjacency Matrix Network with Flexible Loss

In order to capture the potential relationships in the topology space and feature space, and minimize the interference of sampling noise, the proposed CGCN employs a generative adjacency matrix network to fuse node and graph representations to construct a learnable adjacency matrix. Based on this, in pursuit of extracting latent information from the topology space, an autoencoder is employed to explore underlying topology representations from the original graph. Therefore, we adopt an identity matrix \mathbf{I} as the input to the autoencoder, so that the complementary topology adjacency matrix can be defined as $\mathbf{A}^e = g(f(\mathbf{I}))$, where $f(\cdot)$ is an encoder and $g(\cdot)$ is a decoder. With the autoencoder, we can map the identity matrix onto the space with the same dimension as \mathbf{A}^t . Formally, the adjacency matrix $\mathbf{A}_v^e \in \mathbb{R}^{n \times d_v}$ of the v -layer for the autoencoder is

$$\mathbf{A}_v^e = \sigma(\mathbf{A}_{v-1}^e \mathbf{W}^v + \mathbf{b}^v), \quad (4)$$

where $\mathbf{A}_0^e = \mathbf{I}$, $\mathbf{W}^v \in \mathbb{R}^{d_{(v-1)} \times d_v}$ and $\mathbf{b}^v \in \mathbb{R}^{d_v}$ are weight and bias of the v -th layer, respectively. We apply $\sigma(\cdot)$ as the activation function for the autoencoder.

To avoid the interference of sampling noise and useless information in the original data, the autoencoder employs an identity matrix as an input. To extract the implicit topology information associated with \mathbf{A}^t , the loss function of the autoencoder is described as

$$\mathcal{L}_{sa} = \|\mathbf{A}^t - \mathbf{A}^e\|_F^2. \quad (5)$$

To evaluate the quality of the complementary topology matrix \mathbf{A}^e , a new flexible criterion as supervision signals is considered to measure the valid topology representation implied in \mathbf{A}^e . Due to the variation in different datasets, the criterion is desired to be learnable. Therefore, we use a fully-connected neural network as an evaluator to assess the quality of \mathbf{A}^e . We regard the complementary topology adjacency matrix as the input of the evaluator and obtain a value that is positively correlated to the quality of the generative matrix \mathbf{A}^e . The quality value is adopted to evaluate the ability of the autoencoder to capture a more discriminative and complementary representation in the original topology graph. A higher score indicates a better quality of the reconstructed matrix. Therefore, the autoencoder under the supervision of the evaluator can extract more discriminative topology data to reconstruct \mathbf{A}^e with a high value. Thus, we hope that the score \mathcal{G}_e obtained from the evaluator meets

$$\mathcal{G}_e = \max_{\mathbf{w}_e, \mathbf{b}_e} (\text{Eval}(\mathbf{A}^e)), \quad (6)$$

where \mathbf{w}_e and \mathbf{b}_e are the weight and bias of the fully-connected neural network. Therefore, $\text{Eval}(\cdot)$ evaluates the quality of the complementary topology information of \mathbf{A}_v^e defined as

$$\mathbf{A}_v^e = \sigma(\mathbf{A}_{v-1}^e \mathbf{W}_e^v + \mathbf{b}_e^v), \quad (7)$$

where \mathbf{A}_v^e is the embedding of the v -th layer for the evaluator. In order to augment the ability of the evaluator to differentiate the quality of data, we measure the quality of the original adjacency matrix \mathbf{A}^t as a high criterion by the evaluator. Meanwhile, to confuse the autoencoder, we input the complementary topology adjacency matrix \mathbf{A}^e reconstructed by the autoencoder as low-score samples into the evaluator. In the meantime, the autoencoder attempts to generate more discriminative and complementary topology information during learning. The aim is to let the autoencoder output a more valid adjacency matrix against the evaluator through the alternating training process

of two networks. Therefore, the objective of the two networks is defined as

$$\mathcal{G}_e = \max_{\mathbf{w}_t, \mathbf{b}_t} (\text{Eval}(\mathbf{A}^t)) + \min_{\mathbf{w}_e, \mathbf{b}_e} (\text{Eval}(\mathbf{A}^e)), \quad (8)$$

where \mathbf{w}_t and \mathbf{b}_t are the weight and bias of evaluator. The above equation indicates that during the training procedure, the original data are scored as higher as possible, and the matrix generated by the autoencoder is scored as lower as possible. Therefore, the evaluator is described as a loss function because it gets a smaller value when receiving an expected \mathbf{A}^e , and outputs a larger value conversely. In addition, we set the score from 0 to 1 to calculate the differences between the original matrix \mathbf{A}^t and the generative matrix \mathbf{A}^e . Thus the loss function of the evaluator is defined as

$$\mathcal{L}_{se} = - [\log (\text{Eval}(\mathbf{A}^t)) + \log (1 - \text{Eval}(\mathbf{A}^e))]. \quad (9)$$

Through the alternative training of the autoencoder and evaluator, the potential topology information is extracted from the topology space. Besides, in order to capture the latent feature information from the nodes, we employ the k NN method to construct a feature adjacency matrix \mathbf{A}^f . Specifically, we first calculate the similarity matrix $\mathbf{C} \in \mathbb{R}^{n \times n}$ of all nodes by

$$\mathbf{C}_{ij} = \frac{\mathbf{x}_i \cdot \mathbf{x}_j^T}{\|\mathbf{x}_i\| \cdot \|\mathbf{x}_j\|}, \quad (10)$$

and then select the top k similar nodes as neighbors of each node. To enable the topology and feature information fusion, we give a weight α to \mathbf{A}^e , and the generative adjacency matrix confused with topology and feature information is updated by

$$\mathbf{A}^p = \alpha \mathbf{A}^e + (1 - \alpha) \mathbf{A}^f. \quad (11)$$

Subsequently, the latent adjacency matrix \mathbf{A}^p combined with feature matrix \mathbf{X} constructs a new graph $\mathbf{G}_p(\mathbf{X}, \mathbf{A}^p)$.

C. GCN with Pseudo-labeling Mechanism

To extract the common information from the two spaces in different views Φ_1 and Φ_2 , we jointly feed the original topology graph $\mathbf{G}_t(\mathbf{X}, \mathbf{A}^t)$ and the learnable topology graph containing feature information $\mathbf{G}_p(\mathbf{X}, \mathbf{A}^p)$ into GCN layers with shared parameters. Then the final output embeddings can be represented as \mathbf{H}^{Φ_1} and \mathbf{H}^{Φ_2} , respectively:

$$\mathbf{H}^{\Phi_1} = \text{softmax} \left(\hat{\mathbf{A}}^t \sigma \left(\hat{\mathbf{A}}^t \mathbf{X} \mathbf{W}^{(0)} \right) \mathbf{W}^{(1)} \right), \quad (12)$$

$$\mathbf{H}^{\Phi_2} = \text{softmax} \left(\mathbf{A}^p \sigma \left(\mathbf{A}^p \mathbf{X} \mathbf{W}^{(0)} \right) \mathbf{W}^{(1)} \right), \quad (13)$$

where $\hat{\mathbf{A}}^t = \tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}}^t \tilde{\mathbf{D}}^{-\frac{1}{2}}$, $\tilde{\mathbf{A}}^t = \mathbf{A}^t + \mathbf{I}$ and $\tilde{\mathbf{D}}_{ii} = \sum_j \tilde{\mathbf{A}}_{ij}^t$. $\mathbf{W}^{(i)}$ ($i \in \{0, 1\}$) denotes a trainable weight matrix.

For making full use of different views of topology graphs and feature graphs, we design a self-supervised strategy to enrich the supervision signals on the basis of the unlabeled set Ψ_u . To ensure the reliability of pseudo information, the proposed method sets the following conditions before generating pseudo labels to guarantee the reliability. First, because the pseudo information is highly relevant to the reliability of GCN model, we train a fixed number of training epochs on

the initial labeled and unlabeled samples to learn more reliable node relationships. Second, after a certain number of training epochs, we set a high threshold δ and select the vertices from the embeddings \mathbf{H}^{Φ_1} and \mathbf{H}^{Φ_2} that satisfy the maximum value $\max_v h_{iv}^{\Phi_t} > \delta$, and set $j = \arg \max_v h_{iv}^{\Phi_t}$ ($t = 1, 2$) as the pseudo label of the i -th vertex. Finally, an alignment mechanism is applied to further guarantee the reliability of chosen data. Therefore, we align the pseudo labels on the same selected nodes from two embeddings to ensure that the same node from different embeddings has the same pseudo label, thereby facilitating the propagation of label information.

These selected nodes are sorted by their pseudo labels and interlinked within the same class distribution. The selected nodes make up the pseudo-labeling set Ψ_p and the row vectors picked from \mathbf{H}^{Φ_1} come into being the pseudo-labeling embedding $\mathbf{H}^p \in \mathbb{R}^{m \times c}$, where m is the number of the selected nodes. Besides, the corresponding pseudo labels construct a pseudo label set $\mathbf{Y}^p \in \mathbb{R}^{m \times c}$. $\mathbf{Y}_{ij}^p = 1$ represents the i -th node belonging to j -th pseudo label. We create a pseudo adjacency matrix $\mathbf{A}^s \in \mathbb{R}^{n \times n}$ with nodes of the same category as 1 and 0 otherwise. In order to fully exploit the credibility of the pseudo labels, we take \mathbf{A}^s as the feedback to refine the generative latent adjacency matrix \mathbf{A}^p . Given a weight λ on \mathbf{A}^s , this progress is formulated as

$$\mathbf{A}^p = \mathbf{A}^p + \lambda \mathbf{A}^s. \quad (14)$$

In order to leverage more high-confident pseudo labeled data on graphs with limited labels, we also consider the selected pseudo-labeling information in the semi-supervised loss to maximize the consistency of the two distinct graphs, thereby promoting the dissemination of label information. Detail description is shown as follows.

D. Semi-supervised Contrastive Loss

In the proposed module, we improve the semi-supervised contrastive loss to maximize the consistency between topology and feature graphs. Specially, we add the pseudo-labeling technique to design a semi-supervised contrastive loss, which consists of two components: the supervised and unsupervised contrastive losses.

Contrastive learning is expected to make the embeddings of the same node from different views more similar, and outputs of different nodes from distinct views are more dissimilar. Therefore, we describe the unsupervised contrastive loss as

$$\mathcal{L}_{uc} = \frac{1}{2n} \sum_{i=1}^n (\mathcal{L}_{uc}^{\Phi_1}(\mathbf{x}_i) + \mathcal{L}_{uc}^{\Phi_2}(\mathbf{x}_i)), \quad (15)$$

where $\mathcal{L}_{uc}^{\Phi_1}(\mathbf{x}_i)$ and $\mathcal{L}_{uc}^{\Phi_2}(\mathbf{x}_i)$ denote the unsupervised contrastive losses of views Φ_1 and Φ_2 , respectively. Specifically, $\mathcal{L}_{uc}^{\Phi_1}(\mathbf{x}_i)$ is measured by

$$\mathcal{L}_{uc}^{\Phi_1}(\mathbf{x}_i) = - \log \frac{\exp \left(\langle \mathbf{h}_i^{\Phi_1}, \mathbf{h}_i^{\Phi_2} \rangle \right)}{\sum_{j=1}^n \exp \left(\langle \mathbf{h}_i^{\Phi_1}, \mathbf{h}_j^{\Phi_2} \rangle \right)}, \quad (16)$$

where $\mathbf{h}_k^{\Phi_i}$ defines the k -th row vector of the output embedding \mathbf{H}^{Φ_i} ($i \in \{0, 1\}$), and $\langle \cdot, \cdot \rangle$ denotes the inner product mea-

suring the similarity of $\mathbf{h}_k^{\Phi_1}$ and $\mathbf{h}_l^{\Phi_2}$. $\mathcal{L}_{uc}^{\Phi_2}(\mathbf{x}_i)$ is calculated as

$$\mathcal{L}_{uc}^{\Phi_2}(\mathbf{x}_i) = -\log \frac{\exp(\langle \mathbf{h}_i^{\Phi_2}, \mathbf{h}_i^{\Phi_1} \rangle)}{\sum_{j=1}^n \exp(\langle \mathbf{h}_i^{\Phi_2}, \mathbf{h}_j^{\Phi_1} \rangle)}. \quad (17)$$

It is worth noting that a lower $\mathcal{L}_{uc}^{\Phi_1}$ (or $\mathcal{L}_{uc}^{\Phi_2}$) indicates a higher similarity. Furthermore, in order to incorporate the scarce and reliable labels, and take advantage of unlabeled information, a new supervised contrastive loss blended with pseudo labels is defined as

$$\mathcal{L}_{sc} = \frac{1}{2(l+m)} \sum_{i=1}^{l+m} (\mathcal{L}_{sc}^{\Phi_1}(\mathbf{x}_i) + \mathcal{L}_{sc}^{\Phi_2}(\mathbf{x}_i)). \quad (18)$$

Generally, the supervised contrastive losses of \mathbf{x}_i in Ψ_l and Ψ_p can be calculated by

$$\mathcal{L}_{sc}^{\Phi_1}(\mathbf{x}_i) = -\log \frac{\sum_{k=1}^{l+m} \mathbb{1}_{[r_i^{\Phi_1} = r_k^{\Phi_2}]} \exp(\langle \mathbf{h}_i^{\Phi_1}, \mathbf{h}_k^{\Phi_2} \rangle)}{\sum_{j=1}^{l+m} \exp(\langle \mathbf{h}_i^{\Phi_1}, \mathbf{h}_j^{\Phi_2} \rangle)}, \quad (19)$$

$$\mathcal{L}_{sc}^{\Phi_2}(\mathbf{x}_i) = -\log \frac{\sum_{k=1}^{l+m} \mathbb{1}_{[r_i^{\Phi_2} = r_k^{\Phi_1}]} \exp(\langle \mathbf{h}_i^{\Phi_2}, \mathbf{h}_k^{\Phi_1} \rangle)}{\sum_{j=1}^{l+m} \exp(\langle \mathbf{h}_i^{\Phi_2}, \mathbf{h}_j^{\Phi_1} \rangle)}, \quad (20)$$

where $\mathbf{h}_r^{\Phi_1}$ and $\mathbf{h}_r^{\Phi_2}$ rely on \mathbf{H}^{Φ_1} , \mathbf{H}^{Φ_2} and \mathbf{H}^p in both the labeled set Ψ_l and the pseudo set Ψ_p .

By combining the unsupervised contrastive loss and the supervised contrastive loss, we obtain the semi-supervised contrastive loss with pseudo labels:

$$\mathcal{L}_{ssc} = \mathcal{L}_{sc} + \eta \mathcal{L}_{uc}, \quad (21)$$

where η represents a weight of \mathcal{L}_{sc} . Consequently, we leverage the scarce yet relatively reliable unlabeled information to provide additional supervised signals.

E. Model Training

To obtain an overall predictive output \mathbf{O} , we combine the embeddings \mathbf{H}^{Φ_1} and \mathbf{H}^{Φ_2} for capturing information on feature space and topology space represented as

$$\mathbf{O} = \beta \mathbf{H}^{\Phi_1} + (1 - \beta) \mathbf{H}^{\Phi_2}, \quad (22)$$

where $\beta \in (0, 1)$ is a hyperparameter. In addition, cross-entropy loss is used to evaluate the differences between the predictive output and the ground truth as

$$\mathcal{L}_{en} = - \sum_{i=1}^l \sum_{j=1}^c \mathbf{Y}_{ij}^l \ln \mathbf{O}_{ij}, \quad (23)$$

where \mathbf{Y}^l is the labeled matrix. Therefore, the overall loss function of the proposed method can be described as

$$\mathcal{L} = \gamma \mathcal{L}_{en} + (1 - \gamma) \mathcal{L}_{ssc}, \quad (24)$$

where γ is a hyperparameter to regulate the importance between the semi-supervised contrastive loss \mathcal{L}_{ssc} and cross-entropy loss \mathcal{L}_{en} . In a summary, the procedure for CGCN is presented in Algorithm 1.

Algorithm 1 Contrastive Graph Convolutional Networks with generative adjacency matrix (CGCN)

Input: Graph $G_t(\mathbf{X}, \mathbf{A}^t)$, label set \mathbf{Y} , threshold δ , training interval Γ and the number of iterations ξ_1 and ξ_2 .
Output: Node embedding \mathbf{O} .

- 1: # Pre-train the complementary adjacency matrix \mathbf{A}^e ;
- 2: **while** not convergent or reaching ξ_1 **do**
- 3: Forward propagation with an identity matrix as an input in Eq. (4), compute \mathcal{L}_{sa} and optimize the autoencoder by backward propagation with Eq. (5);
- 4: Forward propagation with an identity matrix as an input in Eq. (4), compute \mathcal{L}_{se} and optimize the evaluator by backward propagation with Eq. (9);
- 5: **end while**
- 6: Obtain a complementary topology matrix \mathbf{A}^e ;
- 7: Generate a feature matrix \mathbf{A}^f by kNN method in Eq. (10);
- 8: Construct a latent adjacency matrix \mathbf{A}^p by confusing \mathbf{A}^e and \mathbf{A}^f using Eq. (11);
- 9: **while** not convergent or reaching ξ_2 **do**
- 10: Compute \mathbf{H}^{Φ_1} and \mathbf{H}^{Φ_2} according to Eqs. (12) and (13);
- 11: **if** epochs % $\Gamma == 0$ **then**
- 12: Iterate over the row vectors from \mathbf{H}^{Φ_1} and \mathbf{H}^{Φ_2} ;
- 13: Select the i -th node satisfying $\max_v h_{iv}^{\Phi_t} > \delta$ and set $j = \arg \max_v h_{iv}^{\Phi_t}$ ($t = 1, 2$) as pseudo labels;
- 14: Use the vectors of selected nodes from \mathbf{H}^{Φ_1} to form an embedding \mathbf{H}^p , and the pseudo labels build \mathbf{Y}^p ;
- 15: Construct a pseudo adjacency matrix \mathbf{A}^s with neighbors belonging to the same pseudo labels;
- 16: Refine the latent adjacency matrix \mathbf{A}^p by Eq. (14);
- 17: **end if**
- 18: Calculate \mathcal{L}_{ssc} and \mathcal{L}_{en} using Eqs. (21), (23) and update trainable parameters with back propagation in Eq. (24);
- 19: **end while**
- 20: **return** Node embedding \mathbf{O} .

IV. EXPERIMENTS

A. Experimental Setup

1). **Datasets.** In this subsection, we present an overview of real-world datasets in our experiments, which contain six different types of data so that the effectiveness of CGCN framework can be measured from different fields. The six real-world graph datasets are shown in Table III.

- **Flickr:** This dataset is constructed by forming links between shared Flickr public images, where nodes represent users and edges stand for their relationships, which are formed from the same location, gallery or collection, etc.
- **ACM:** It is a dataset where nodes represent papers and edges express that two papers have the same author. It can be used to conduct citation networks, paper content, and other data integration studies.
- **BlogCatalog:** It is a social network constructed from the BlogCatalog website, where nodes are made up of the keywords of user profiles, and the labels represent the topic categories provided by the authors.
- **CoraFull:** CoraFull is a citation network repository, where nodes represent papers and edges stand for their

TABLE II: Node classification performance (ACC% and F1%) on six datasets.

Datasets	Metrics	L/C	GCN	GAN-GCN	GAT	FA-GCN	Scatter-GCN	AM-GCN	CG3-GCN	CGCN
Flickr	ACC	20	54.30	73.20	43.10	60.11	48.35	<u>74.28</u>	63.50	81.10
		40	67.00	76.70	47.30	68.53	59.50	<u>78.92</u>	65.30	83.20
		60	69.80	77.50	51.36	71.05	64.51	<u>81.06</u>	66.30	83.60
	F1	20	52.41	73.16	39.94	59.65	45.94	<u>73.83</u>	22.62	80.73
		40	66.19	76.30	45.38	68.86	57.98	<u>78.57</u>	23.03	82.77
		60	69.25	77.33	49.40	71.52	63.47	<u>80.76</u>	23.33	83.36
ACM	ACC	20	88.00	<u>93.40</u>	84.70	87.38	90.14	90.56	90.10	98.50
		40	91.20	<u>94.70</u>	89.16	90.27	91.43	90.60	91.30	98.90
		60	92.20	<u>95.20</u>	89.58	91.93	91.47	90.26	91.70	99.10
	F1	20	88.01	<u>93.30</u>	84.75	87.38	90.18	90.64	47.15	98.49
		40	88.50	<u>94.63</u>	89.08	90.21	91.46	90.65	46.58	98.88
		60	92.23	<u>95.15</u>	89.71	91.93	91.53	90.31	46.71	99.08
BlogCatalog	ACC	20	<u>85.70</u>	79.50	60.54	82.30	70.44	82.58	32.20	91.50
		40	<u>88.50</u>	82.70	63.14	81.72	75.18	85.38	85.30	91.90
		60	<u>89.30</u>	84.20	67.32	84.77	79.23	86.56	83.70	92.50
	F1	20	<u>85.03</u>	79.05	59.26	77.16	69.03	81.94	31.78	91.26
		40	<u>88.01</u>	82.46	62.35	81.08	77.22	85.09	31.16	91.79
		60	<u>88.77</u>	84.02	66.75	84.33	78.59	86.30	31.76	92.33
CoraFull	ACC	20	<u>57.44</u>	51.55	57.44	57.30	54.67	54.76	26.00	57.70
		40	<u>60.63</u>	51.07	58.39	60.01	59.74	60.10	27.90	60.67
		60	63.20	51.53	62.19	63.33	<u>64.03</u>	63.75	35.40	65.18
	F1	20	<u>57.04</u>	51.07	56.68	56.10	46.82	51.05	11.63	57.13
		40	58.38	48.55	56.89	<u>58.61</u>	48.84	52.12	12.48	59.39
		60	<u>60.48</u>	48.99	58.62	59.33	50.09	52.32	18.91	61.42
Film	ACC	20	25.10	26.00	19.30	<u>29.29</u>	24.29	26.34	22.00	29.70
		40	26.21	27.50	21.00	<u>29.95</u>	25.40	26.70	19.90	30.40
		60	27.80	28.50	22.18	<u>30.27</u>	25.63	27.54	22.19	32.10
	F1	20	24.48	<u>25.43</u>	19.18	24.23	23.06	17.95	20.33	28.15
		40	25.89	<u>27.09</u>	20.66	24.70	24.45	20.95	21.24	29.84
		60	27.10	<u>28.58</u>	21.90	26.18	24.57	21.39	21.08	31.19
Citeseer	ACC	20	70.30	64.90	68.32	67.23	68.42	<u>71.22</u>	69.00	72.60
		40	71.61	69.50	71.24	71.30	70.26	<u>73.40</u>	72.00	74.70
		60	72.86	69.90	71.30	72.33	71.25	<u>74.20</u>	73.00	74.90
	F1	20	<u>67.50</u>	60.88	65.89	63.67	66.07	66.26	44.02	69.33
		40	<u>68.83</u>	65.10	68.54	68.19	66.86	67.84	44.47	70.41
		60	<u>70.01</u>	64.66	68.75	68.95	68.11	68.06	44.83	70.49

TABLE III: Statistical summary of test datasets.

Datasets	# Nodes	# Edges	# Features	# Classes
Flickr	7,575	239,738	12,047	9
ACM	3,025	13,128	1,870	3
BlogCatalog	5,196	171,743	8,189	6
CoraFull	19,793	65,311	8,710	70
Film	7,600	15,009	932	5
Citseer	3,327	4,732	3,703	6

citations. The nodes are labeled based on the paper topics. It consists of 70 class distributions.

- **Film:** It is a film social network that describes the relationships between films, and it contains 7,600 nodes

and 171,743 edges with 5 class distributions.

- **Citeseer:** This is a research paper citation network, where nodes represent publications and edges stand for citation links. Each link is described by a word vector that indicates the presence or absence of the corresponding word in the dictionary.
- 2). *Baselines.* We compare our proposed method with the following state-of-the-art methods.
 - **GCN** [9]: It is a semi-supervised graph convolutional network which learns node representations by aggregating information from neighbors.
 - **GAN-GCN:** Instead of a traditional topology graph, we use the complementary topology graph calculated by an autoencoder with feature information as the input of GCN

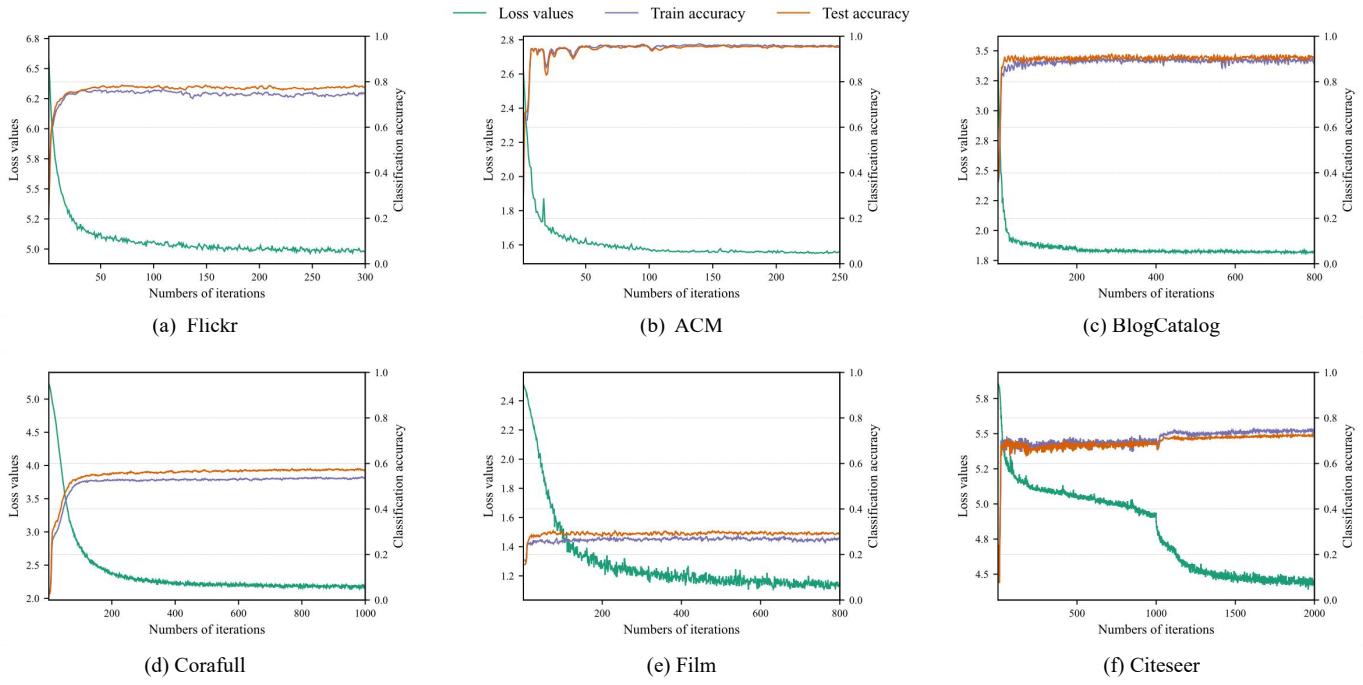


Fig. 2: Convergence curves of training loss values, validation accuracy and test accuracy with CGCN on six datasets.

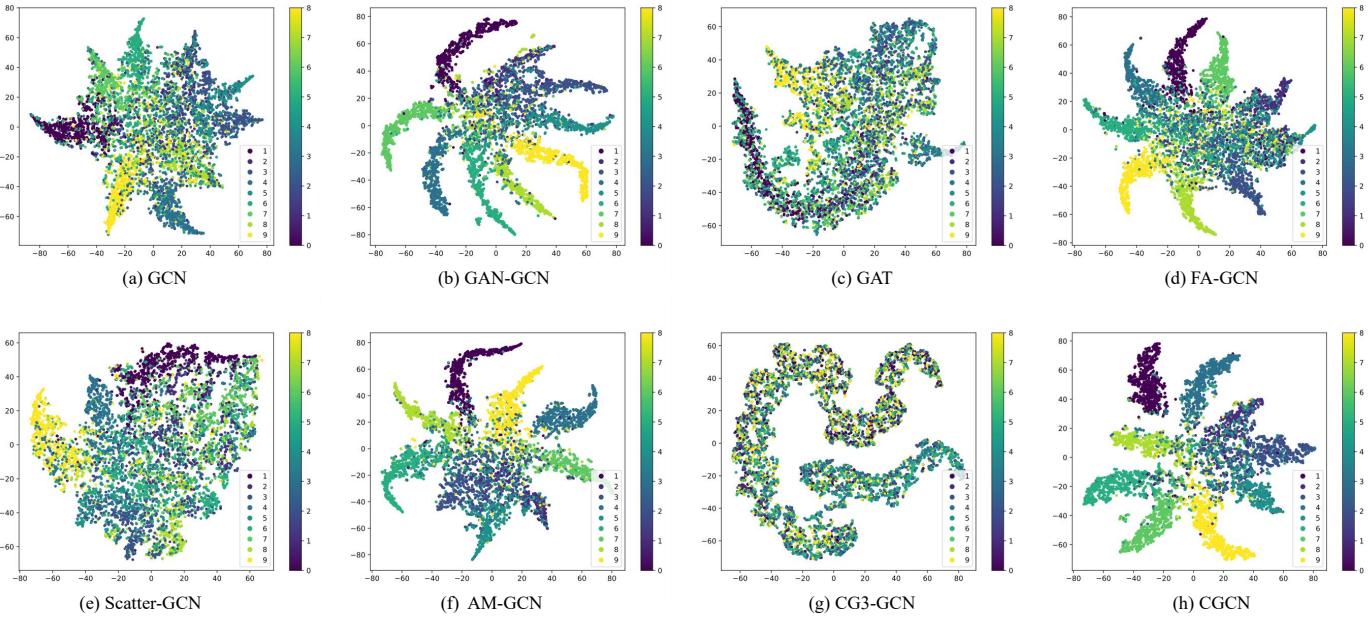


Fig. 3: T-SNE visualization of node classification results of compared methods on Flickr.

for comparison.

- **GAT** [47]: It is a convolution-style neural network that uses an attention mechanism to propagate node relationships and learn node features.
- **FA-GCN** [19]: It is a frequency adaptation graph convolutional network which combines the low-frequency and high-frequency signals for adapting to different tasks.
- **Scatter-GCN** [20]: Scatter-GCN is a semi-supervised GCN-based framework which introduces neural pathways that encode higher-order regularity on graphs.
- **AM-GCN** [27]: AM-GCN is a multi-channel model

which is able to learn suitable importance weights when fusing topology and node feature information.

- **CG3-GCN** [28]: It is a GCN-based approach that designs a semi-supervised contrastive loss and employs data similarities to learn a transductive representation.

3). *Parameter Setting*. In our experiments, all parameter settings of baselines are suggested by their papers. In order to fully evaluate our model, we select three label rates (20/40/60 labels per class) for the training set, and randomly choose 500 and 1,000 nodes as the validation set and the test set, respectively. In the proposed model, the complementary

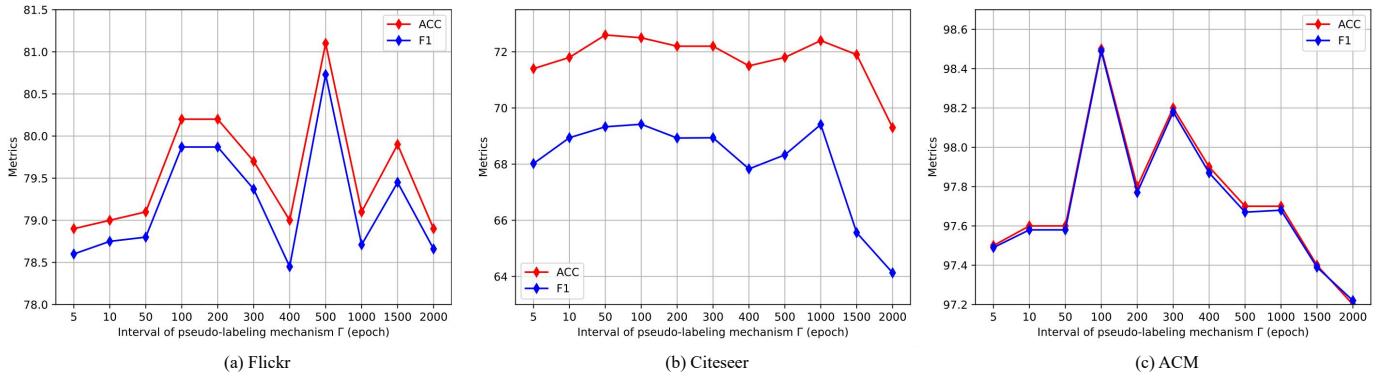


Fig. 4: Influence of interval Γ of pseudo-labeling mechanism on Flickr, Citeseer and ACM.

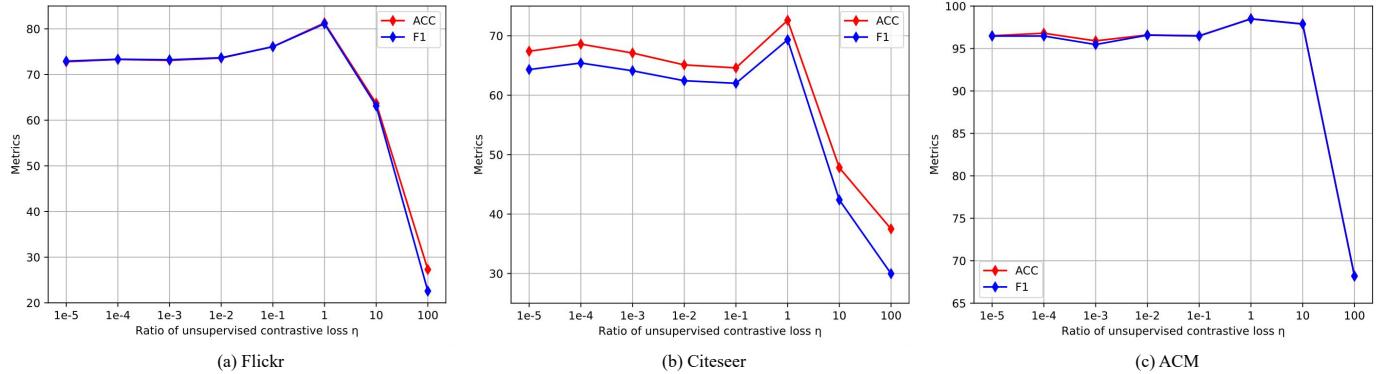


Fig. 5: Parameter sensitivity (ACC% and F1%) of the proposed method w.r.t. η on Flickr, Citeseer and ACM.

topology adjacency matrix \mathbf{A}^e is constructed by an autoencoder of 4 fully-connected layers with the neuron numbers of encoders being $(n, 256, 512, 1024)$ and decoders being $(1024, 512, 256, n)$. Besides, the evaluator also adopts 4 fully-connected layers to assess the quality of the generative matrix, where the hidden neuron numbers are $(1024, 512, 128, 1)$. The Sigmoid function is adopted as the neuron activation of the last layer, and the rest layers use the ReLU function for the autoencoder and evaluator. To extract feature information, we apply k NN method to selecting the top 20 neighbors for each node on the feature matrix, thereby constructing a feature adjacency matrix \mathbf{A}^f . We adopt a 2-layer GCN with ReLU function as the neuron activation of the first layer and Softmax function of the last layer, where the dropout rate l is 0.5 and the weight decay is 5×10^{-4} . We fix the parameter η as 1. We employ Adam optimizer to update learnable parameters with learning rate $lr = 1 \times 10^{-2}$ and weight decay as 5×10^{-4} for the autoencoder, evaluator and GCN model. In addition, we select the confidence threshold as $\delta = 0.99$ and $\eta = 1$.

The proposed CGCN framework is implemented with an early-stop mechanism to avoid overfitting. Two well-known metrics including accuracy (ACC) and macro F1-score (F1) are employed for the performance evaluation.

B. Node Classification Results

We show the experimental results on six real-world datasets in this subsection. The performance of all compared methods

with distinct numbers of labeled data is reported in Table II. From experimental results, we draw some beneficial observations. First, compared with these baselines, CGCN generally obtains the best performance on all datasets. Especially, our method achieves a great improvement on Flickr, ACM, and BlogCatalog datasets. These experimental results verify the effectiveness of the proposed method. In addition, we observe that the accuracy of GAN-GCN is better than that of GCN on some datasets such as Flickr, Film, and ACM, which implies that the representation capacity of the complementary graph we constructed is more powerful than the natural topology graph on specific scenes. Moreover, it also indicates that the relationships combining topology and node spaces are often more discriminative than those in the original topology space. Consequently, a well-constructed joint framework with feature and topology graphs can facilitate the ability of mining robust and generalized representations.

Fig. 2 exhibits the training loss values and the accuracy values of the validation set and test set on six datasets during the training of the proposed algorithm. From the figure, it can be seen that the training loss generally declines within 500 iterations, then it converges slightly on most datasets except Film and Citeseer. The training loss on Film drops continuously and finally becomes convergent, while the loss of Citeseer drops considerably around 1,000 iterations because of the pseudo-labeling mechanism. Besides, we can also observe that the validation accuracy and test accuracy rise as the

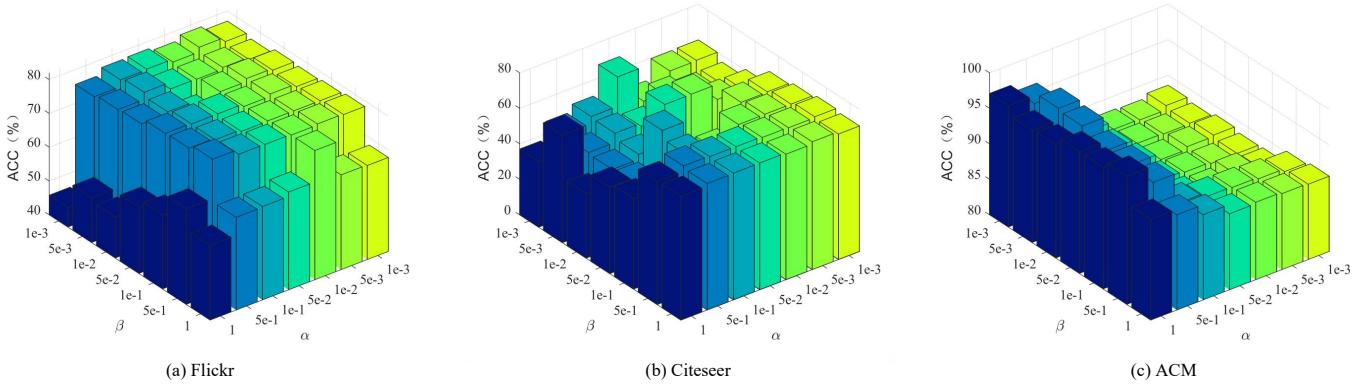


Fig. 6: Parameter sensitivity (ACC%) of the proposed method w.r.t. α and β on Flickr, Citeseer and ACM.

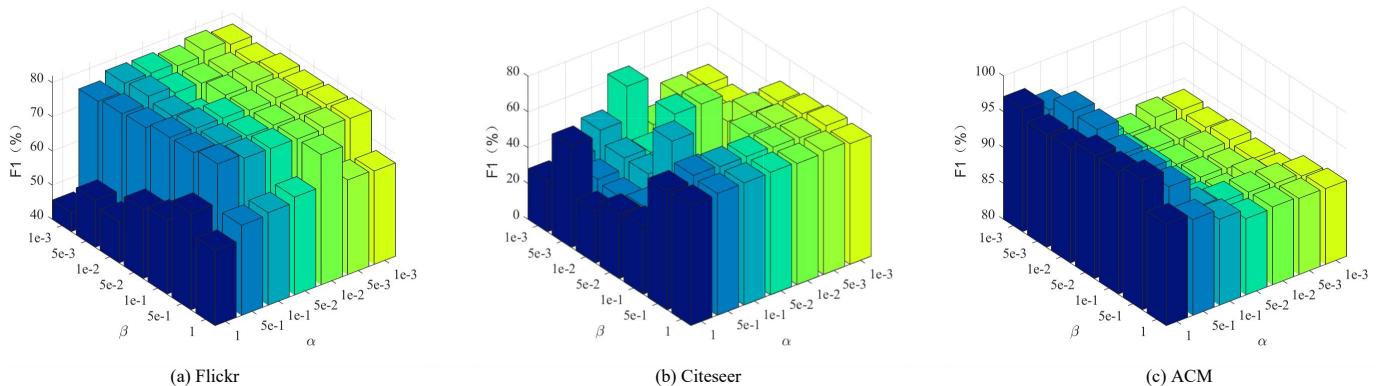


Fig. 7: Parameter sensitivity (F1%) of the proposed method w.r.t. α and β on Flickr, Citeseer and ACM.

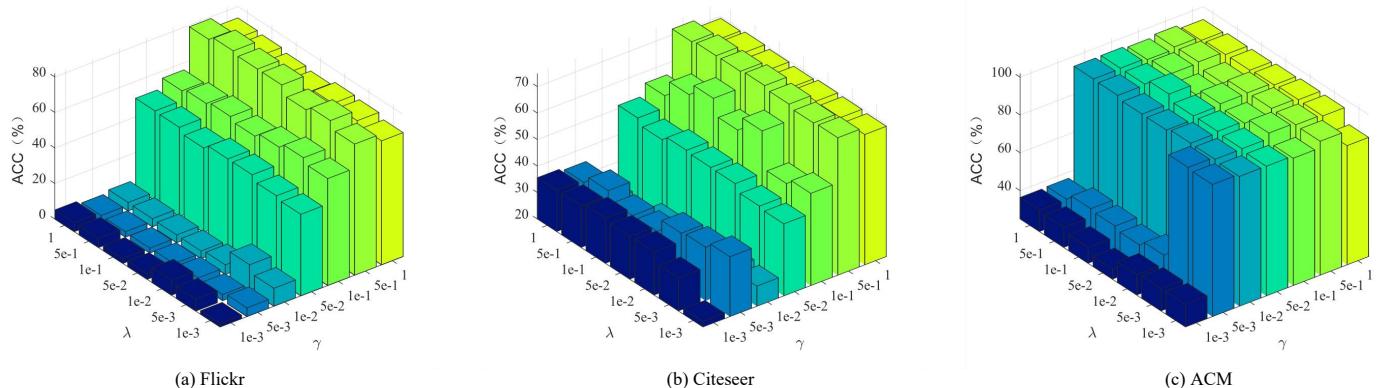


Fig. 8: Parameter sensitivity (ACC%) of the proposed method w.r.t. λ and γ on Flickr, Citeseer and ACM.

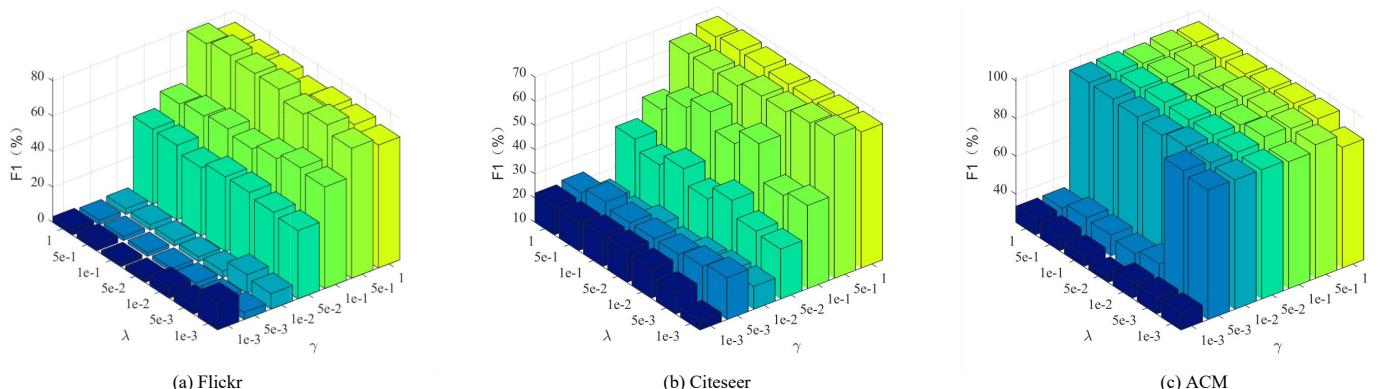


Fig. 9: Parameter sensitivity (F1%) of the proposed method w.r.t. λ and γ on Flickr, Citeseer and ACM.

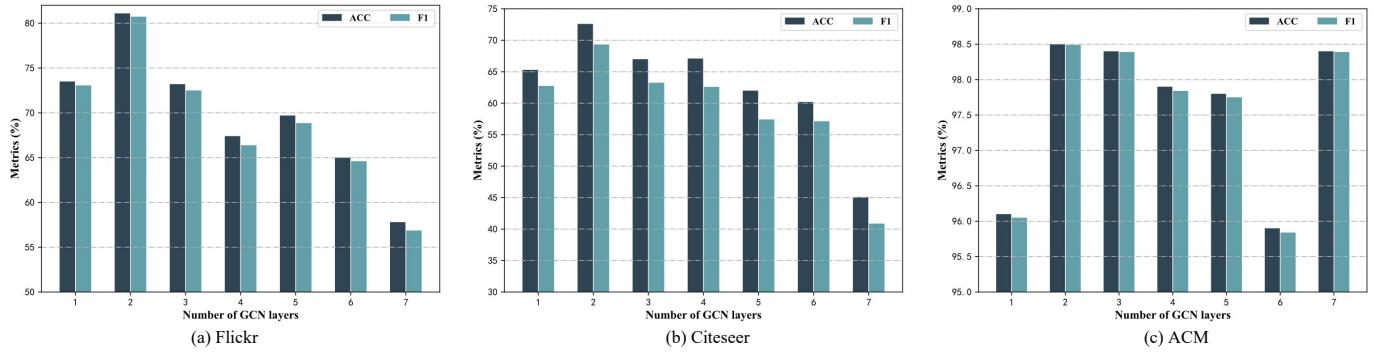


Fig. 10: Influence of the proposed method with the varying numbers of layers on Flickr, Citeseer and ACM.

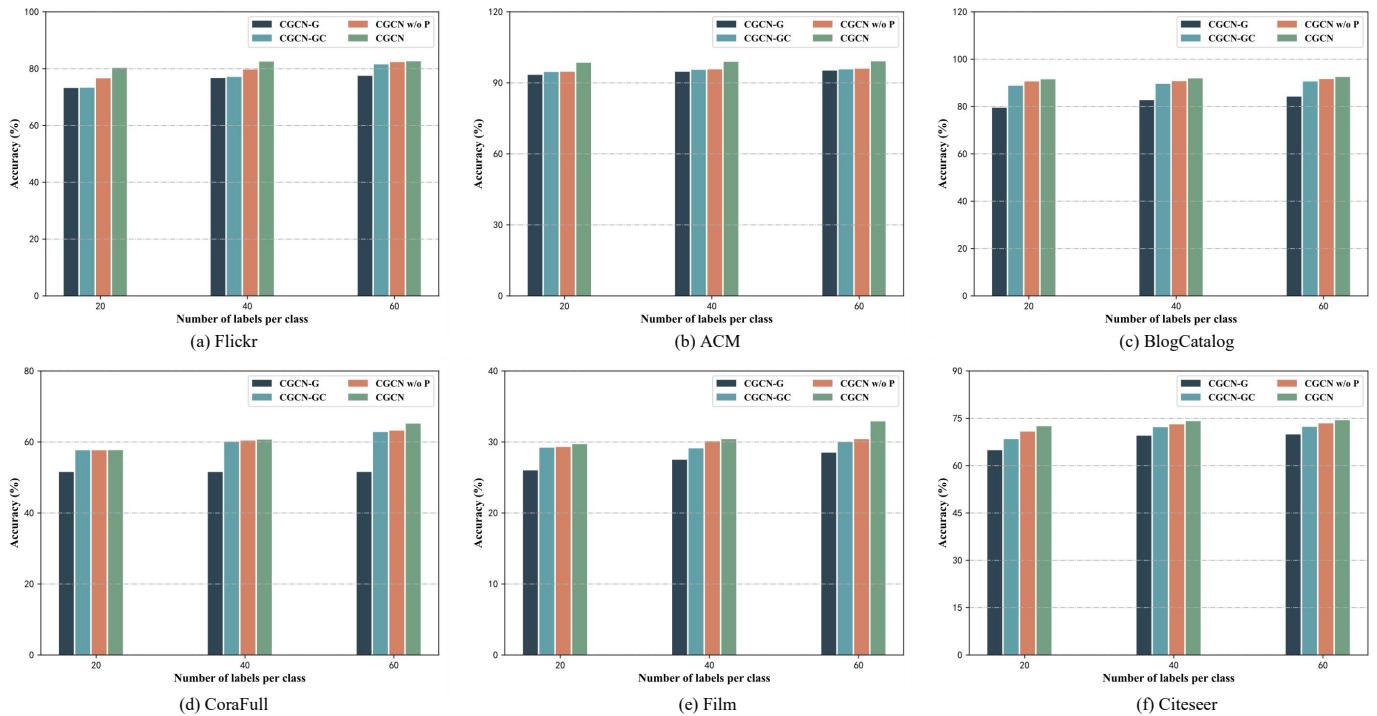


Fig. 11: Ablation study of the proposed model on six datasets with ACC (%).

training loss goes down after 1,000 iterations in Fig. 2 (a). One reasonable explanation is that the proposed pseudo-labeling mechanism improves the robustness of the CGCN model, which has an encouraging effect on Citeseer dataset. The validation accuracy and test accuracy also improve as the training loss decreases on these real-world datasets.

To intuitively demonstrate the differences between CGCN and other state-of-the-art GCN-based methods on classification performance, Fig. 3 employs t-SNE to visualize the classification results of all algorithms on Flickr. The figure indicates that CGCN with suitable hyperparameter settings assigns more accurate class labels, which reveals that the framework of CGCN achieves leading performance in terms of node classification and further validates its superiority compared with other baselines.

C. Parameter Sensitivity Analyses

In this subsection, we conduct parameter sensitivity analyses with ACC and F1 on ACM, Citeseer, and Flickr with

20 labeled samples per class to investigate the performance variations of CGCN under different settings.

In order to observe the effect of the pseudo-labeling mechanism, we study the performance of CGCN by enabling the pseudo-labeling mechanism after every certain amount of epochs Γ in Fig. 4. Besides, we range the value of Γ from 5 to 2,000. The reliability of the generated pseudo labels depends on the stability of the proposed model. For ACM, the performance of CGCN is better when applying pseudo-labeling information and refining the generating adjacency matrix every 50 epochs. It is because that the pseudo-labeling information generated in the early stage is trustworthy and helpful to improve the accuracy of the CGCN framework. For Flickr and Citeseer, the accuracy increases when enabling the pseudo-labeling mechanism every 500 to 1,000 epochs. This suggests that more training epochs are required to stabilize the model and then produce more reliable pseudo information for larger-scale datasets.

Fig. 5 presents the effect of the proportion of the unsu-

ervised contrastive loss η ranging in $[1 \times 10^{-5}, 100]$. It can be seen that the performance rises slowly as the ratio of η gradually increases to 1, and then the performance degrades. This suggests that both the supervised contrastive loss and the unsupervised contrastive loss leave comparable effects on the proposed framework.

Next, we test the effect of α and β in $[1 \times 10^{-3}, 1]$. Fig. 6 and Fig. 7 present the accuracy and F1-score results on three datasets. From the figures, we can observe that the selection of these two hyperparameters has a noticeable influence on classification performance. In general, the optimal performance is achieved when β is fixed as a relatively small value although the value of α varies largely. It validates the effectiveness of the proposed topology adjacency matrix A^e , which indicates that the deeper implicit topology information is helpful for the improvement of the original data. Besides, the classification accuracy is robust to these hyperparameters when α is in a suitable range value on most datasets, indicating that the complementary topology information has a greater effect on small-scale datasets.

In order to check the impact of the weights λ and γ ranging in $[1 \times 10^{-3}, 1]$, we inspect the accuracy and F1-score of the proposed CGCN in Fig. 8 and Fig. 9. From the observation, it can be seen that small λ and γ often lead to poor accuracy in most cases, especially significant for γ . Moreover, it achieves better performance when a suitable γ is selected on Flickr and ACM, which indicates that both the cross-entropy loss and the semi-supervised contrastive loss have a positive impact. On Flickr and Citeseer datasets, the accuracy fluctuates marginally when $\gamma < 0.1$, indicating that choosing parameters tailored for data is critical. Especially, satisfactory performance is gained when the values of λ and γ are 0.5.

We also explore the influence of the varying numbers of GCN layers in Fig. 10. From the figure, we can observe that the model achieves the best performance with a two-layer model. When only one layer is utilized, information propagation may be insufficient owing to the shallow network and inadequate parameters. Besides, it often causes over-smoothing for a multi-layer model, thereby leading to performance degradation.

D. Ablation Study

In order to validate the contribution of the proposed modules, we test the classification accuracy of CGCN with its variants in Fig. 11. Our model is divided into three modules: the self-adaptive adjacency matrix module, the contrastive loss module, and the pseudo-labeling mechanism module. Because the pseudo-labeling mechanism is used to refine the generative adjacency matrix and input semi-supervised contrastive loss, we use the superposition method for ablation.

Especially, CGCN-G represents the semi-supervised graph convolutional network with the generative topology matrix. CGCN-GC denotes that the original topology matrix and the generative topology matrix are concurrently put into the GCN model with shared parameters. CGCN w/o P is CGCN without the pseudo-labeling mechanism and CGCN is the complete model that contains the self-adaptive adjacency ma-

trix, the pseudo-labeling mechanism, and the semi-supervised contrastive loss.

From Fig. 11, it is apparent that the classification accuracy goes up when we stack modules one by one, which suggests that each component of CGCN makes considerable contributions to boosting the performance of the node classification task. Further, it also discloses that compared with the information in the topology space, the learned graph representations combining the feature space and topology space are more discriminative in semi-supervised node classification. Moreover, experimental results also validate the effectiveness of our modules. Therefore, the observation draws a conclusion that the three components of CGCN are beneficial to the performance promotion of semi-supervised classification tasks.

V. CONCLUSION

In this paper, we proposed a contrastive graph neural network framework with a generative adjacency matrix which considered the correlation between nodes and features. For better capturing the complementary topology information and fusing the node features, we designed a learnable self-adaptive adjacency matrix refined by reliable pseudo information. In addition, we utilized a joint self-supervised framework with two mutually supervised GCNs to propagate graph embeddings across both topology and feature graphs, where the semi-supervised contrastive loss was adopted to maximize the consistency between the two networks. We also proposed another pseudo-labeling strategy to enrich the supervision information by generating reliable pseudo labels under the mutual supervision of two networks. Experimental results on benchmark datasets clearly demonstrated the superiority of the proposed framework compared with other state-of-the-art methods in semi-supervised classification tasks.

REFERENCES

- [1] R. Levie, F. Monti, X. Bresson, and M. M. Bronstein, "Cayleynets: graph convolutional neural networks with complex rational spectral filters," *IEEE Transactions on Signal Processing*, vol. 67, pp. 97–109, 2019.
- [2] F. Feng, X. He, H. Zhang, and T. S. Chua, "Cross-gcn: enhancing graph convolutional network with k-order feature interactions," *IEEE Transactions on Knowledge and Data Engineering*, 2021. doi:10.1109/TKDE.2021.3077524.
- [3] X. Jiang, Z. Yang, P. Wen, L. Su, and Q. Huang, "A sparse-motif ensemble graph convolutional network against over-smoothing," in *Proceedings of the 31st International Joint Conference on Artificial Intelligence*, pp. 2094–2100, 2022.
- [4] P. W. Battaglia, R. Pascanu, M. Lai, D. J. Rezende, and K. Kavukcuoglu, "Interaction networks for learning about objects, relations and physics," in *Proceedings of the 30th Conference on Neural Information Processing Systems*, pp. 4502–4510, 2016.
- [5] Y. Ma, S. Wang, C. C. Aggarwal, and J. Tang, "Graph convolutional networks with eigenpooling," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 723–731, 2019.
- [6] X. Wei, R. Yu, and J. Sun, "View-gcn: view-based graph convolutional network for 3d shape analysis," in *Proceedings of the 30th IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1850–1859, 2020.
- [7] Y. Bi, A. Chadha, A. Abbas, E. Bourtsoulatze, and Y. Andreopoulos, "Graph-based spatio-temporal feature learning for neuromorphic vision sensing," *IEEE Transactions on Image Processing*, vol. 29, pp. 9084–9098, 2020.

- [8] S. Wang, Z. Chen, S. Du, and Z. Lin, "Learning deep sparse regularizers with applications to multi-view clustering and semi-supervised classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 5042–5055, 2022.
- [9] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proceedings of the 5th International Conference on Learning Representations*, pp. 1–14, 2017.
- [10] J. Chen, H. He, F. Wu, and J. Wang, "Topology-aware correlations between relations for inductive link prediction in knowledge graphs," in *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, pp. 6271–6278, 2021.
- [11] Y. Shen, W. Dai, C. Li, J. Zou, and H. Xiong, "Multi-scale graph convolutional network with spectral graph wavelet frame," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 7, pp. 595–610, 2021.
- [12] Y. Bi, A. Chadha, A. Abbas, E. Bourtsoulatze, and Y. Andreopoulos, "Graph-based spatio-temporal feature learning for neuromorphic vision sensing," *IEEE Transactions on Image Processing*, vol. 29, pp. 9084–9098, 2020.
- [13] K. Cheng, Y. Zhang, C. Cao, L. Shi, J. Cheng, and H. Lu, "Decoupling gcn with dropgraph module for skeleton-based action recognition," in *Proceedings of the 16th Conference on European Conference on Computer Vision*, pp. 536–553, 2020.
- [14] L. Lei, T. Chen, S. Li, and J. Li, "Micro-expression recognition based on facial graph representation learning and facial action unit fusion," in *Proceedings of the 35th IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1571–1580, 2021.
- [15] Z. Wang, L. Zheng, Y. Li, and S. Wang, "Linkage based face clustering via graph convolution network," in *Proceedings of the 29th IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1117–1125, 2019.
- [16] S. Du, Z. Liu, Z. Chen, W. Yang, and S. Wang, "Differentiable bi-sparse multi-view co-clustering," *IEEE Transactions on Signal Processing*, vol. 69, pp. 4623–4636, 2021.
- [17] S. Huang, Y. Zhang, L. Fu, and S. Wang, "Learnable multi-view matrix factorization with graph embedding and flexible loss," *IEEE Transactions on Multimedia*, 2022. doi:10.1109/TMM.2022.3157997.
- [18] S. Wang, X. Lin, Z. Fang, S. Du, and G. Xiao, "Contrastive consensus graph learning for multi-view clustering," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, pp. 2027–2030, 2022.
- [19] D. Bo, X. Wang, C. Shi, and H. Shen, "Beyond low-frequency information in graph convolutional networks," in *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, pp. 3950–3957, 2021.
- [20] Y. Min, F. Wenkel, and G. Wolf, "Scattering gcn: overcoming over-smoothness in graph convolutional networks," *Proceedings of the 34th Conference on Neural Information Processing Systems*, pp. 14498–14508, 2020.
- [21] S. Abu-El-Haija, B. Perozzi, A. Kapoor, N. Alipourfard, K. Lerman, H. Harutyunyan, G. V. Steeg, and A. Galstyan, "Mixhop: higher-order graph convolutional architectures via sparsified neighborhood mixing," in *Proceedings of the 36th International Conference on Machine Learning*, pp. 21–29, 2019.
- [22] W. Chen, L. Tian, B. Chen, L. Dai, Z. Duan, and M. Zhou, "Deep variational graph convolutional recurrent network for multivariate time series anomaly detection," in *Proceedings of the 39th International Conference on Machine Learning*, pp. 3621–3633, 2022.
- [23] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of the 37th International Conference on Machine Learning*, pp. 1597–1607, 2020.
- [24] X. Xia, H. Yin, J. Yu, Q. Wang, L. Cui, and X. Zhang, "Self-supervised hypergraph convolutional networks for session-based recommendation," in *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, pp. 4503–4511, 2021.
- [25] K. Sun, Z. Lin, and Z. Zhu, "Multi-stage self-supervised learning for graph convolutional networks on graphs with few labeled nodes," in *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, pp. 5892–5899, 2020.
- [26] Q. Li, Z. Han, and X. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pp. 3538–3545, 2018.
- [27] X. Wang, M. Zhu, D. Bo, P. Cui, C. Shi, and J. Pei, "Am-gcn: adaptive multi-channel graph convolutional networks," in *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1243–1253, 2020.
- [28] S. Wan, S. Pan, J. Yang, and C. Gong, "Contrastive and generative graph convolutional networks for graph-based semi-supervised learning," in *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, pp. 10049–10057, 2021.
- [29] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proceedings of the 16th IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1735–1742, 2006.
- [30] C. Gong, D. Tao, K. Fu, and J. Yang, "Fick's law assisted propagation for semisupervised learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, pp. 2148–2162, 2015.
- [31] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," in *Proceedings of the 2nd International Conference on Learning Representations*, pp. 1–14, 2014.
- [32] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proceedings of the 30th Conference on Neural Information Processing Systems*, pp. 3837–3845, 2016.
- [33] K. Liu, L. Gao, N. M. Khan, L. Qi, and L. Guan, "A multi-stream graph convolutional networks-hidden conditional random field model for skeleton-based action recognition," *IEEE Transactions on Multimedia*, vol. 23, pp. 64–76, 2021.
- [34] X. Xu, T. Wang, Y. Yang, A. Hanjalic, and H. T. Shen, "Radial graph convolutional network for visual question generation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, pp. 1654–1667, 2021.
- [35] M. Lei, P. Quan, R. Ma, Y. Shi, and L. Niu, "Diggcn: learning compact graph convolutional networks via diffusion aggregation," *IEEE Transactions on Cybernetics*, vol. 52, pp. 912–924, 2022.
- [36] M. A. T. Figueiredo and A. K. Jain, "Unsupervised learning of finite mixture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 381–396, 2002.
- [37] T. Li, K. Zhang, S. Shen, B. Liu, Q. Liu, and Z. Li, "Image co-saliency detection and instance co-segmentation using attention graph clustering based graph convolutional network," *IEEE Transactions on Multimedia*, vol. 24, pp. 492–505, 2021.
- [38] Z. Wang, M. Eisen, and A. Ribeiro, "Learning decentralized wireless resource allocations with graph neural networks," *IEEE Transactions on Signal Processing*, vol. 70, pp. 1850–1863, 2022.
- [39] Q. Li, X. Zhang, H. Liu, Q. Dai, and X.-M. Wu, "Dimensionwise separable 2-d graph convolution for unsupervised and semi-supervised learning on graphs," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 953–963, 2021.
- [40] J. Aneja, A. Schwing, J. Kautz, and A. Vahdat, "A contrastive learning approach for training variational autoencoder priors," in *Proceedings of the 35th Conference on Neural Information Processing Systems*, pp. 480–493, 2021.
- [41] K. He, H. Fan, Y. Wu, S. Xie, and R. B. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the 30th IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9726–9735, 2020.
- [42] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," in *Proceedings of the 34th Conference on Neural Information Processing Systems*, pp. 1–13, 2020.
- [43] K. Hassani and A. H. K. Ahmadi, "Contrastive multi-view representation learning on graphs," in *Proceedings of the 37th International Conference on Machine Learning*, pp. 4116–4126, 2020.
- [44] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1225–1234, 2016.
- [45] W. Yu, C. Zheng, W. Cheng, C. C. Aggarwal, D. Song, B. Zong, H. Chen, and W. Wang, "Learning deep network representations with adversarially regularized autoencoders," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 2663–2671, 2018.
- [46] K. Tu, P. Cui, X. Wang, P. S. Yu, and W. Zhu, "Deep recursive network embedding with regular equivalence," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2357–2366, 2018.
- [47] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proceedings of the 6th International Conference on Learning Representations*, pp. 1–12, 2018.



Luying Zhong received her B.S. degree from the College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China in 2021. She is currently pursuing the M.S. degree with the College of Computer and Data Science, Fuzhou University, Fuzhou, China. Her current research interests include graph learning, contrastive learning, graph neural networks and deep learning.



Jinbin Yang received his B.S. degree from the College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China in 2021. He is currently pursuing the M.S. degree with the College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China. His current research interests include deep clustering, deep learning and graph neural networks.



Zhaoliang Chen received his B.S. degree from the College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China in 2019. He is currently pursuing the Ph.D. degree with the College of Computer and Data Science, Fuzhou University, Fuzhou, China. He is also currently visiting Faculty of Computer Science, University of Vienna, Vienna, Austria. His current research interests include feature selection, matrix factorization, graph neural networks and deep learning.



Shiping Wang received the Ph.D. degree from the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China, in 2014. He was a Research Fellow with Nanyang Technological University from August 2015 to August 2016. He is currently a Full Professor and Qishan Scholar with the College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China. His research interests include machine learning, data mining, computer vision, optimization theory and granular computing.