Learnable Graph Convolutional Network With Semisupervised Graph Information Bottleneck

Luying Zhong, Zhaoliang Chen, Zhihao Wu, Shide Du, Zheyi Chen, *Associate Member, IEEE*, and Shiping Wang, *Senior Member, IEEE*

***Abstract*— Graph convolutional network (GCN) has gained widespread attention in semisupervised classification tasks. Recent studies show that GCN-based methods have achieved decent performance in numerous fields. However, most of the existing methods generally adopted a fixed graph that cannot dynamically capture both local and global relationships. This is because the hidden and important relationships may not be directed exhibited in the fixed structure, causing the degraded performance of semisupervised classification tasks. Moreover, the missing and noisy data yielded by the fixed graph may result in wrong connections, thereby disturbing the representation learning process. To cope with these issues, this article proposes a learnable GCN-based framework, aiming to obtain the optimal graph structures by jointly integrating graph learning and feature propagation in a unified network. Besides, to capture the optimal graph representations, this article designs dual-GCN-based meta-channels to simultaneously explore local and global relations during the training process. To minimize the interference of the noisy data, a semisupervised graph information bottleneck (SGIB) is introduced to conduct the graph structural learning (GSL) for acquiring the minimal sufficient representations. Concretely, SGIB aims to maximize the mutual information of both the same and different meta-channels by designing the constraints between them, thereby improving the node classification performance in the downstream tasks. Extensive experimental results on real-world datasets demonstrate the robustness of the proposed model, which outperforms state-of-the-art methods with fixed-structure graphs.**

***Index Terms*— Graph convolutional network (GCN), graph information bottleneck, graph learning, mutual information, semisupervised learning.**

I. Introduction

With the powerful capability of expression, graphs have been universally leveraged to depict real-world applications, that is, knowledge graphs [1], [2], social connections [3], [4], paper citations [5], [6], [7], and affinity networks [8], [9], [10]. Recently, the graph convolutional network (GCN) [11] has drawn considerable attention owing to the desirable performance for graph learning. Based on the critical theoretical foundations, the GCN efficiently employs the message-passing mechanism to aggregate feature information from neighbors through their connective relationships to obtain a meaningful node-level representation for the classification tasks. Numerous variants of the GCN have been widely proposed in different fields, such as computer vision [12], [13], [14], social network analysis [15], [16], [17], node clustering [18], [19], [20], and node classification [21], [22], [23]. A large number of studies have validated that GCN-based models can attain inspiring performance in various tasks.

Considering that labeling data is still time-consuming and labor-intensive, semisupervised learning [24], [25], [26] with GCNs which endeavors to utilize limited labeled data as well as relatively large amounts of unlabeled data for model training has been researched. Although these GCN-based semisupervised methods have gained noticeable progress and demonstrated their superior performance in recent years, they generally used a fixed graph with noises that may not be optimal for node classification. In other words, only using the original topology pattern containing noisy data for feature propagation, while ignoring the dynamical potential global connective relationships conducted by the training process may lose some hidden connections, thereby leading to incomplete node representations and undesired performance. There are still limited studies making efforts to learn dynamical graphs combined with the GCN. For instance, Jiang et al. [27] proposed a graph learning convolutional framework that designed a conductive graph learning operation to learn an optimal representation for serving semisupervised classification tasks. Nevertheless, it did not consider the interference of noises involved in data, and missing or wrong connections may propagate unsatisfactory node information when aggregating features by connective relationships. In summary, to make feature propagation more efficient, it is vital to explore graph structural learning (GSL) while alleviating the disruption of noises as much as possible.

In particular, the information bottleneck is a critical principle that endeavors to learn a robust representation from noisy data. It encourages learning a minimum sufficient representation of noisy data to limit the disruption of noises and preserve the maximal valid information on the target through mutual information. Prior works have applied the information

Manuscript received 9 March 2023; revised 19 September 2023; accepted 29 September 2023. This work was supported in part by the National Natural Science Foundation of China under Grant U21A20472 and Grant 62276065 and in part by the National Key Research and Development Plan of China under Grant 2021YFB3600503. *(Corresponding author: Shiping Wang.)*

The authors are with the College of Computer and Data Science and the Fujian Provincial Key Laboratory of Network Computing and Intelligent Information Processing, Fuzhou University, Fuzhou 350116, China (e-mail: luyingzhongfzu@163.com; chenzl23@outlook.com; zhihaowu1999@gmail.com; dushidems@gmail.com; z.chen@fzu.edu.cn; shipingwangphd@163.com).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TNNLS.2023.3322739.

Digital Object Identifier 10.1109/TNNLS.2023.3322739

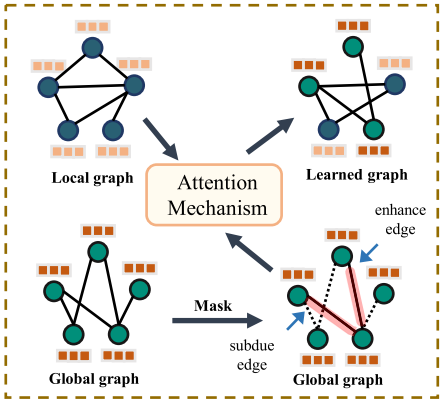

Fig. 1. Graph structural construction based on local and global connective relationships. For a global graph, only edges (blue) with higher importance are enhanced, and other edges (alternatives) are subdued. Then combined with local and global graphs, an attention mechanism endeavors to learn the importance of different nodes, thereby forming a learned graph.

bottleneck [28], [29], [30] to representation learning. However, these methods only applied the information bottleneck to fixed-structural graph data and did not focus on the part of learning structures, which may lead to poor correlations for the final predictive representations. In addition, they did not consider the consistency between different channels, which may make different channels unbending for classification tasks.

To address these issues, this article proposes a flexible GCN-based framework named learnable GCN with semisupervised graph information bottleneck (LGCN-SGIB), which jointly learns dynamical graph structures with local and global connective relationships and designs a semisupervised graph information bottleneck (SGIB) to alleviate the interference of noises. To fully explore the connective relationships, we design dual-GCN-based meta-channels to extract optimal graph structures from topology and feature spaces, respectively. Each meta-channel contains three fundamental modules. Specifically, the global exploration (GE) module aims to collect global hidden connections according to the supervision of the training process. The GSL module endeavors to update graph structures based on local and global information. The Graph convolution (GCNconv) module is to learn a node-level representation from learnable graph structures and features for the downstream task. To better clarify the GSL, Fig. 1 intuitively exhibits its details. We call the initial adjacency matrix and the adjacency matrix learned by the GE module the local graph and global graph, respectively. Considering that the global graph may contain undesired connections between nodes, a Mask$(\cdot)$ function, which adaptively refines the edges during the model training, is employed to further adjust the adjacency matrix. Then an attention mechanism is developed to fuse local and global information to obtain a more comprehensive graph. Furthermore, on the basis of LGCN-SGIB, we design an SGIB as an optimization objective function, which constrains the mutual information between inputs, node embeddings, and predictive representations to suppress the data noises and maximize the consistency of effective information, thus guiding the whole framework. The main contributions of this article are summarized as follows.

1) Propose a learnable dual-meta-channels-based GCN framework to address graph data that contain noises from topology and feature spaces, which considers both local and global neighbor connections.
2) Design a GSL strategy to dynamically update graphs and integrate it with GCN for representation learning.
3) A differentiable SGIB is proposed, which alleviates the interference of noises from original data and enhances the correlations between meta-channels.
4) The proposed approach is applied to semisupervised node classification tasks, and substantial experiments on benchmark datasets demonstrate the superiority of LGCN-SGIB compared with the state-of-the-arts.

The rest of this article is organized as follows. Related works on the GCN and information bottleneck are reviewed in Section II. Section III elaborates the proposed LGCN-SGIB. Section IV verifies the proposed framework via substantial experiments. Conclusions and future work are presented in Section V.

II. Related Work

In this section, some existing studies relevant to our work are reviewed, starting with an overview of the GCN and then focusing on the introduction of the information bottleneck.

A. Semisupervised Learning With the GCN

Semisupervised learning [31] has been an attractive topic recently. The operating mechanism of semisupervised learning endeavors to leverage limited labeled signals with massive unlabeled samples to guide the model training. Semisupervised learning has been universally applied in various domains. For example, Guan et al. [32] introduced a semisupervised metapath-guided framework to solve the attribute entities problem for personalized compatibility modeling. Luo et al. [33] developed a semisupervised feature analysis method to learn robust features for recognition tasks. Guan et al. [34] proposed a bi-directional heterogeneous graph hash framework to improve efficiency for recommendation tasks. Among them, graph-based semisupervised learning has gained widespread attention due to its remarkable expressive capabilities.

Based on this, as a powerful tool of graph-based semisupervised learning, the GCN is remarkably promising in graph neural networks. Generally, each layer of the propagation rule in the GCN is formulated as

$$\text{GCNconv}_{\Theta}(\mathbf{A}, \mathbf{X}) = \sigma(\mathbf{A}\mathbf{X}\Theta) \tag{1}$$

where Θ is a trainable parameter of the GCNconv module, σ is a nonlinear activation function, and $\mathbf{A}$ is the renormalized adjacency matrix. The GCN is capable of aggregating feature information from neighbors, which facilitates learning a discriminative representation for specific tasks. Owing to the powerful effectiveness of the GCN, numerous extensions and variants of the GCN [35], [36], [37], [38] have achieved optimal performance for semisupervised classification tasks. A key idea of GCNs is exploring the consistency of both

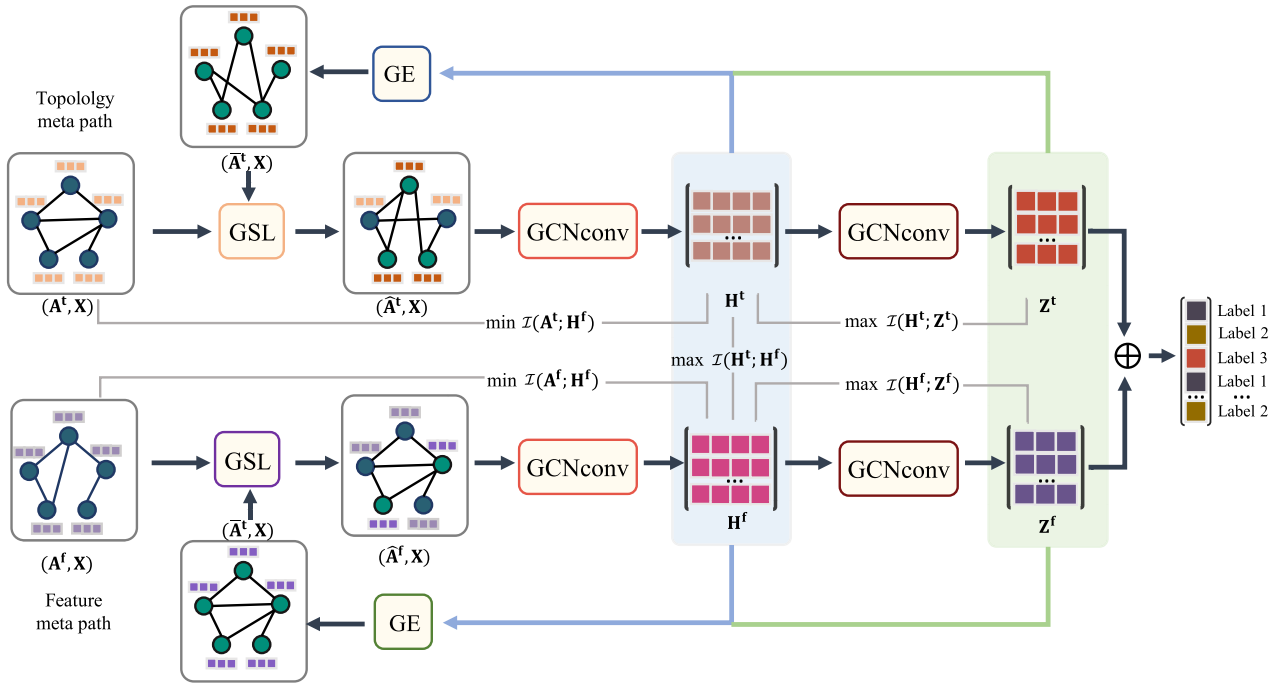

Fig. 2. Architecture of the proposed framework, which aims to solve semisupervised node classification problems. Concretely, as a supervised signal, the GE module is employed to explore connective relationships in the global space. The GSL module endeavors to activate the relations in global learning and refine the graph structure on local and global information. After that, GCNconv modules are utilized to learn node-level representations combined with the SGIB to constrain mutual information for classification tasks.

feature and topology graph channels to enhance the expressivity of node embeddings. For instance, Xu et al. [39] developed a dual-GCN-based deep feature aggregation path framework to extract high spatial information for classification tasks. Wang et al. [40] employed two-channel GCNs to explore potential information from feature and topology spaces in semisupervised node classification tasks. Some state-of-the-arts have focused on structural learning to improve the capabilities of GCNs to discriminate between nodes on graphs. For example, Min et al. [41] presented a semisupervised GCN-based framework that applied residual convolutions and scattered transformations to enhance higher-order regularity on graphs. Wu et al. [42] simplified the GCN by precomputing a high-order weighted matrix and removing the nonlinear activation function to approximate the propagation of multilayer GCNs. Several works have attempted to utilize self-supervised learning to solve the limitation of insufficient labeled signals. Wan et al. [29] employed a semisupervised loss on the GCN-based framework to enrich the potential information from unlabeled samples. Sun et al. [43] utilized a multistage GCN framework that generated extra pseudo-supervised information in the downstream task to alleviate the limitation of supervised signals.

Although these GCN-based semisupervised methods have gained remarkable achievements, they still suffer from the shortage of using fixed graphs with noises, which may cause undesired feature propagation and could not dynamically handle global information produced by the training process, resulting in suboptimal prediction classification. Since it is essential to learn optimal graph structures [44] for semisupervised node classification tasks, it is still a great challenge to adaptively explore GSL and alleviate the disruption of noises during model training.

B. Information Bottleneck

Information bottleneck is a critical principle in information theory, which aims to employ mutual information as the regularizer to balance between the original data and generalization. The goal of the information bottleneck encourages that the representation obtained from the original data contains the maximal valid information for the target and excludes the additional information that is irrelevant for prediction tasks. To this end, the information bottleneck directly minimizes the mutual information between the original data $\mathbf{x}$ and the obtained representation $\mathbf{h}$ and maximizes the mutual information between $\mathbf{h}$ and the predictive representation $\mathbf{y}$ simultaneously, formulated as

$$\mathcal{L}_{IB} = \mathcal{I}(\mathbf{y};\mathbf{h}) - \beta\mathcal{I}(\mathbf{x};\mathbf{h}) \tag{2}$$

where $\mathcal{I}(\cdot;\cdot)$ denotes the mutual information, β is a Lagrangian parameter, and $\mathbf{x}$, $\mathbf{h}$, $\mathbf{y}$ denote vectors.

The information bottleneck has received significant attention in machine learning and deep learning. Recently, some research studies have been proposed to integrate the information bottleneck principle into structural learning to alleviate the disruption of noises. Alemi et al. [28] designed a variational estimation by approximating the distribution using a neural network to the mutual information. Wu et al. [25] introduced the information bottleneck for graph learning to capture the minimal sufficient information from graph-structured data. Yu et al. [45] solved a subgraph recognition problem by

estimating the mutual information in irregular graph data. Yang et al. [46] employed the information bottleneck for the heterogeneous graph neural network for semisupervised classification tasks. Sun et al. [30] advanced the information bottleneck principle that optimized both the graph structure and the graph representation to facilitate training stability and efficiency.

Nevertheless, although these approaches have successfully employed the information bottleneck for various applications, they did not consider evaluating the consistency of diverse channels, which may lead to weak correlations between channels, thereby obtaining undesired performance.

III. Proposed Method

A. Overview and Notations

At the beginning of this section, some basic notations used in this article are clarified. Given graph data, the node feature matrix is denoted as $\mathbf{X} = [\mathbf{x}_1; \mathbf{x}_2, \ldots, \mathbf{x}_n] \in \mathbb{R}^{n\times d}$, where $\mathbf{x}_i$ is the feature vector of the ith node, n is the number of samples, and d is the feature dimension. Specifically, there are l labeled nodes, and the remaining $u = n - l$ samples are unlabeled with $l \ll u$. The adjacency matrix is denoted as $\mathbf{A}^t \in \mathbb{R}^{n\times n}$, with $\mathbf{A}^t_{ij} = 1$ representing that there exists a connection between the ith node and the jth node and 0 otherwise. $\tilde{\mathbf{A}}^t$ is a renormalized matrix. $\mathbf{Y} \in \mathbb{R}^{n\times c}$ is a label matrix, with c being the number of class and $\mathbf{Y}_{ij} = 1$ representing the ith node belongs to the jth class. Given the graph representation $\mathcal{G}_t(\mathbf{A}^t, \mathbf{X})$, this article mainly focuses on dynamically extracting the local and global connective relationships by decreasing the interference of noisy information to obtain optimal graph structures, which are further used to extract the node-level representation $\mathbf{Z} \in \mathbb{R}^{n\times c}$ for the semisupervised node classification tasks.

B. Learnable GCN

To obtain optimal graph structures and better apply them to the node classification tasks, a learnable GCN-based framework that jointly explores local and global connections, as well as alleviates the noises hidden in data is proposed. As shown in Fig. 2, the whole framework contains three primary modules: the GE module, the GSL module, and the GCNcnov module. Concretely, as a supervised signal, the GE module utilizes the k-means clustering method combined with a screening mechanism to generate connections with global information. The GSL module aims to refine graph structures on local and global information during the inference process. The GCNcnov module is employed to learn the node-level representation according to the connective pattern and node embeddings. In addition, to fully explore the correlated information, dual meta-channels are designed to extract the optimal topology graph structure and the feature graph structure from topology and node features, respectively. Each meta-channel includes a GE module, a GSL module, and two GCNcnov modules, and an optimization objective function is designed to guide the whole training by combining the SGIB. To better explain the proposed LGCN-SGIB, we first give its overall process, then we elaborated GE module in Section III-C, GSL module in Section III-D, SGIB in Section III-E, and the module training in Section III-F.

The latent feature information $\mathbf{A}^f$ is extracted from the raw features by k-nearest neighbor algorithm (kNN), and, therefore, the (i, j)th element of $\mathbf{A}^f$ is denoted as

$$\mathbf{A}^f_{ij} = \begin{cases} 1, & \mathbf{x}_j \in k\text{NN}(\mathbf{x}_i) \text{ or } \mathbf{x}_j \in k\text{NN}(\mathbf{x}_i) \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where $k\text{NN}(\mathbf{x}_i)$ is the set of the k nearest neighbors of $\mathbf{x}_i$. Then, we renormalize $\tilde{\mathbf{A}}^f = \tilde{\mathbf{D}}^{-(1/2)}(\mathbf{I} + \mathbf{A}^f)\tilde{\mathbf{D}}^{-(1/2)}$, where $\tilde{\mathbf{D}}_{ii} = \sum_j \tilde{\mathbf{A}}^f_{ij} \in \mathbb{R}^{n\times n}$ is the degree matrix derived from the self-connection matrix $\tilde{\mathbf{A}}^f$. Then, the feature adjacency matrix $\mathbf{A}^f$ and the raw features $\mathbf{X}$ form the feature graph $\mathcal{G}_f(\mathbf{A}^f, \mathbf{X})$. The whole procedure is shown below.

Given $\mathcal{G}_t(\mathbf{A}^t, \mathbf{X})$ and $\mathcal{G}_f(\mathbf{A}^f, \mathbf{X})$, we employ two GCNconv modules. Specifically, taking the topology metapath as an example, the node embedding of the first GCNconv module is defined as

$$\mathbf{H}^t = \text{GCNconv}_{\Theta_1}(\hat{\mathbf{A}}^t, \mathbf{X}) \tag{4}$$

where Θ_1 is the parameter of the GCNconv module to extract the common information from the topology space.

To obtain the final predictive representations, the second GCNconv module is employed with the parameter Θ_2, denoted as

$$\mathbf{Z}^t = \text{GCNconv}_{\Theta_2}(\hat{\mathbf{A}}^t, \mathbf{H}^t). \tag{5}$$

It is noted that both the topology and feature metapaths share the same parameters in GCNconv modules. Besides, $\hat{\mathbf{A}}^t = \tilde{\mathbf{A}}^t$ and $\hat{\mathbf{A}}^f = \tilde{\mathbf{A}}^f$ when GCNconv modules are in the first iteration. Then, we employ a GE module to explore the global relationship between topology and feature spaces, defined as

$$\bar{\mathbf{A}}^* = \text{GE}\big(\mathbf{H}^*, \mathbf{Z}^*\big), \quad * \in \{t, f\}. \tag{6}$$

Finally, we utilize a GSL model to obtain the learned topology graph and the learned feature graph, formulated as

$$\hat{\mathbf{A}}^* = \text{GSL}_{\Theta_*}\big(\bar{\mathbf{A}}^*, \mathbf{H}^*, \mathbf{Z}^*\big), \quad * \in \{t, f\} \tag{7}$$

where Θ_* is the trainable parameter of the GSL module. The final node predictive representation is defined as

$$\mathbf{Z} = \mathbf{Z}^t + \alpha\mathbf{Z}^f. \tag{8}$$

Furthermore, an SGIB module is employed to conduct the whole learnable GCN framework by acquiring the minimal sufficient graph structures through constraining the mutual information of both the same meta-channel and different meta-channels, thereby decreasing the interference of noises and optimizing the consistency of homogeneous representations. Specific details are elaborated as follows.

C. Global Exploration

As a supervised signal, the GE module aims to generate global connective relationships according to the supervision of both the node embeddings and the predictive representations. Therefore, before exploring global information, we execute the k-means method on node embeddings to obtain global relationships on topology and feature spaces and explore the weak labels of each node to further screen nodes for constructing the global graphs. Taking topology meta-channel

as an example, we first cluster all nodes into distinct categories based on the geometric criterion using the k-means algorithm and learn a centroid matrix $\bar{\mathbf{C}}^t \in \mathbb{R}^{c\times m}$ as

$$\bar{\mathbf{C}}^t = k\text{means}(\mathbf{H}^t). \tag{9}$$

For each of the lth clusters in the topology meta-channel, the computation of the weak label of the cluster is

$$\bar{y}_l^t = \arg\min_i \left\| \bar{\mathbf{c}}_l^t - \mathbf{c}_i^t \right\|^2 \tag{10}$$

where $\bar{\mathbf{c}}_l^t$ is the centroid of cluster l, $\mathbf{c}_i^t$ is the centroid in labeled data with $i \in [c]$, and $\bar{y}_l^t$ is the weak labels of nodes belonging to the cluster l forming weak label set $\bar{\mathbf{Y}}^t$.

1) Screening Mechanism: The target of the screening mechanism is to filter out more reliable nodes to construct global graphs according to the obtained weak labels and the predictive labels from $\mathbf{Z}^t$ and $\mathbf{Z}^f$. We screen the more reliable nodes by aligning the weak label $\bar{\mathbf{Y}}^t$ for the topology meta-channel and the predictive labels of the output layer for the topology meta-channel to maximize the consistency between meta-channels, that is,

$$\hat{\mathbf{X}}_t = \text{Append}\Big(\mathbb{1}_{[\bar{y}_i^t = \hat{y}_i^t]}\mathbf{x}_i\Big), \quad i \in [n] \tag{11}$$

where Append$(\cdot)$ function expends the node $\mathbf{x}_i$ to set $\hat{\mathbf{X}}_t$ and the $\mathbb{1}_{[\cdot]}$ is an indicator function that equals to 1 if satisfying the condition of $[\cdot]$, and 0 otherwise. $\hat{\mathbf{X}}_t$ denotes a node set of where the nodes meet $[\cdot]$, and $\bar{y}_p^t$ and $\hat{y}_p^t = \arg\max_j \mathbf{z}_p^t$ are the weak labels and predictive label of the pth node, respectively. Then, a global topology graph $\bar{\mathbf{A}}^t$ is built by $\hat{\mathbf{X}}_t$, where the nodes with the same weak labels are connected when the value is equal to 1. As for the feature meta-channel, the global feature graph $\bar{\mathbf{A}}^f$ is also created like the above process.

D. Graph Structural Learning

Although graphs $\bar{\mathbf{A}}^t$ and $\bar{\mathbf{A}}^f$ are built by the filtered reliable nodes, there may be some undesired connections between nodes. Based on this, the goal of the GSL module is to preserve the higher important connections without destroying the graph structures and then fuse the local and global connections. Since it is difficult to evaluate the degree of importance between connections, a natural idea is to adjust these weights of connections adaptively according to the training process. Toward this end, we design a learnable mask function Mask$(\cdot)$ to adjust the connective relation between the ith node and the jth node, denoted as

$$\mathbf{A}^* = \bar{\mathbf{A}}^* \odot \big(\text{ReLU}(\mathbf{S} - \Phi) + \text{ReLU}(\mathbf{S}^\top - \Phi)\big), \quad * \in \{t, f\} \tag{12}$$

where $\odot$ is the Hadamard product, $\mathbf{S} \in \mathbb{R}^{n\times n}$ represents the shared learnable coefficient matrix to control the weight for each edge, and $\Phi \in \mathbb{R}^{n\times n}$ denotes the shared coefficient bias matrix to reduce local noises. To ensure the nonnegativity of the coefficients, Sigmoid function is applied to $\mathbf{S}$, $\mathbf{S}^\top$, and Φ, respectively. Therefore, a connective relationship exists between the ith node and the jth node when the coefficient of the ith node or the coefficient of the jth node is greater than the bias value, thereby achieving the purpose of adaptively selecting the right connections and inhibiting the wrong connections through the model training.

Furthermore, plunging into the attention mechanism to learn the importance of each node for the fusion of local and global connections, the learned graphs are finally defined as

$$\hat{\mathbf{A}}^* = \text{Attention}\big(\tilde{\mathbf{A}}^*, \mathbf{A}^*\big), \quad * \in \{t, f\} \tag{13}$$

where the linear network is adopted for the attention mechanism to learn the importance of nodes.

E. Semisupervised Graph Information Bottleneck

Considering that the graph structures may have noisy data, which may lead to poor performance for the downstream task. Based on this, we extend the information bottleneck that learns minimal sufficient representation using the mutual information to the SGIB that is combined with the limited labeled signals to conduct the proposed model. The target of SGIB is to learn minimal sufficient graph representations by constraining as much irrelevant information to the target task as possible and reserving the maximal useful information from label information. Meanwhile, we also strengthen the consistency between node embeddings from distinct meta-channels by maximizing their mutual information. Hence, the objective function of SGIB can be written as

$$\begin{aligned}\mathcal{L}_S = &\big(\mathcal{I}(\mathbf{A}^t; \mathbf{H}^t) + \mathcal{I}(\mathbf{A}^f; \mathbf{H}^f)\big) - \beta\mathcal{I}(\mathbf{H}^t; \mathbf{H}^f) \\ &- \gamma\big(\mathcal{I}(\mathbf{Z}^t; \mathbf{H}^t) + \mathcal{I}(\mathbf{Z}^f; \mathbf{H}^f)\big)\end{aligned} \tag{14}$$

where the first term aims to minimize the information $\mathbf{H}^t$ (and $\mathbf{H}^f$) from data $\hat{\mathbf{A}}^t$ (and $\hat{\mathbf{A}}^f$), the second term maximizes the embeddings between the distinct meta-channels, while the third term maximizes the mutual information between the node embedding $\mathbf{H}^t$ (and $\mathbf{H}^f$) and the predictive representation $\mathbf{Z}^t$ (and $\mathbf{Z}^f$), where β and γ are hyperparameters to balance all terms.

1) Estimation: To optimize the parameters of the framework, we need to estimate these mutual information. Based on this, for the first term in (14), we transform it into minimizing the upper bound to obtain the minimal sufficient representations. Furthermore, since the node embeddings and the original graph structures have a dependent relationship denoted as $\{\mathbf{A}^t \to \hat{\mathbf{A}}^t \to \mathbf{H}^t\}$ and $\{\mathbf{A}^f \to \hat{\mathbf{A}}^f \to \mathbf{H}^f\}$ according to the Markov chain principle, we have $\mathcal{I}(\mathbf{A}^t; \mathbf{H}^t) \le \mathcal{I}(\mathbf{A}^t; \hat{\mathbf{A}}^t)$. Similarly, we also obtain $\mathcal{I}(\mathbf{A}^f; \mathbf{H}^f) \le \mathcal{I}(\mathbf{A}^f; \hat{\mathbf{A}}^f)$. Based on this, the upper bound is computed as

$$\begin{aligned}\hat{\mathcal{I}}(\mathbf{A}^t; \mathbf{H}^t) \le &\hat{\mathcal{I}}(\mathbf{A}^t, \hat{\mathbf{A}}^t) \\ &= D_{KL}\big(p(\hat{\mathbf{A}}^t|\mathbf{A}^t) \,||\, r(\mathbf{A}^t)\big).\end{aligned} \tag{15}$$

The above specific derivation is provided in the Appendix. To specify the upper bound, we assume that both $p(\hat{\mathbf{A}}^t|\mathbf{A}^t)$ and $r(\mathbf{A}^t)$ are subject to Bernoulli distributions, and a_{ij}^t has the half probability that equals to 0 or 1, defined as

$$\hat{\mathcal{I}}(\mathbf{A}^t; \mathbf{H}^t) \le \sum_{i=1}^{n}\sum_{j=1}^{n} D_{KL}\big(\text{Bern}(\hat{a}_{ij}^t) \,||\text{Bern}(0.5)\big). \tag{16}$$

Analogously, $\hat{\mathcal{I}}(\mathbf{A}^f, \mathbf{H}^f)$ satisfies

$$\hat{\mathcal{I}}(\mathbf{A}^f; \mathbf{H}^f) \leq \sum_{i=1}^{n}\sum_{j=1}^{n} D_{KL}\Big(\text{Bern}(\hat{a}_{ij}^f) \,||\text{Bern}(0.5)\Big). \quad (17)$$

To maximize the mutual information of $\mathcal{I}(\mathbf{H}^t; \mathbf{H}^f)$, we also transform the problem into maximizing the lower bound of $\mathcal{I}(\mathbf{H}^t; \mathbf{H}^f)$. Considering that the labeled information encourages the performance of the node classification task, we propose a method to estimate the lower bound of the mutual information combined with the known labeled signals. Specific theoretical derivation is shown in the Appendix.

Lemma 1: Given a latent variable M and two random variables X_1 and X_2, we have

$$\begin{aligned}\mathcal{I}(X_1; X_2) \geq\, & \mathbb{E}_{q(X_1,X_2|M=1)}\big[\log q(M=1|X_1, X_2)\big] \\ & + \mathbb{E}_{q(X_1,X_2|M=0)}\big[\log q(M=0|X_1, X_2)\big] \quad (18)\end{aligned}$$

where $M = 1$ denotes that (x_1, x_2) belongs to the same class sampling from the joint distribution and $M = 0$ represents that (x_1, x_2) belongs to different classes from the marginal distribution.

Based on this, we apply the above lemma to $(\mathbf{H}^t, \mathbf{H}^f)$. However, considering that the estimation of the distributions $q(M = 1|X_1, X_2)$ and $q(M = 0|X_1, X_2)$ are difficult, so we introduce a bivariate function $g(\mathbf{H}^t, \mathbf{H}^f) \to \{0, 1\}$ to $q(M = 1|X_1, X_2)$. Similarly, $q(M = 0|X_1, X_2)$ is substituted by $(1 - g(\mathbf{H}^t, \mathbf{H}^f)) \to \{0, 1\}$.

Consequently, the right term of (18) is described as

$$\begin{aligned}\hat{\mathcal{I}}(\mathbf{H}^t, \mathbf{H}^f) =\, & \mathbb{E}_{q(\mathbf{H}^t,\mathbf{H}^f|M=1)}\big[\log g\big(\mathbf{H}^t, \mathbf{H}^f\big)\big] \\ & + \mathbb{E}_{q(\mathbf{H}^t,\mathbf{H}^f|M=0)}\big[\log\big(1 - g\big(\mathbf{H}^t, \mathbf{H}^f\big)\big)\big]. \quad (19)\end{aligned}$$

In addition, the bivariate function $g(\mathbf{H}^t, \mathbf{H}^f)$ satisfies

$$\begin{aligned}g\big(\mathbf{H}^t, \mathbf{H}^f\big) =\, & \frac{1}{2l}\sum_{i=1}^{l}\frac{\sum_{j=1}^{l}\mathbb{1}_{[y_i=y_j]}\exp\langle\mathbf{h}_i^t, \mathbf{h}_j^f\rangle}{\sum_{k=1}^{l}\exp\langle\mathbf{h}_i^t, \mathbf{h}_k^f\rangle} \\ & + \frac{1}{2l}\sum_{j=1}^{l}\frac{\sum_{i=1}^{l}\mathbb{1}_{[y_j=y_i]}\exp\langle\mathbf{h}_j^f, \mathbf{h}_i^t\rangle}{\sum_{k=1}^{l}\exp\langle\mathbf{h}_j^f, \mathbf{h}_k^t\rangle} \quad (20)\end{aligned}$$

where $\mathbf{h}_i^*$ denotes the ith vector of $\mathbf{H}^*$, with $* \in [t, f]$ and $\langle\cdot\rangle$ denotes the inner product.

For the third term of (14), we aim to maximize the lower bound of the mutual information $\hat{\mathcal{I}}(\mathbf{Z}^t; \mathbf{H}^t)$ and $\hat{\mathcal{I}}(\mathbf{Z}^f; \mathbf{H}^f)$ instead of computing the minimal $-(\mathcal{I}(\mathbf{Z}^t; \mathbf{H}^t) + \mathcal{I}(\mathbf{Z}^f; \mathbf{H}^f))$. Here, MINE [47] is employed to estimate the lower bound of mutual information as

$$\hat{\mathcal{I}}\big(\mathbf{Z}^t; \mathbf{H}^t\big) = \sup_{\Omega_t}\mathbb{E}_{p(\mathbf{Z}^t,\mathbf{H}^t)}\big[T_{\Omega_t}\big] - \log\big(\mathbb{E}_{p(\mathbf{Z}^t)p(\mathbf{H}^t)}\big[e^{T_{\Omega_t}}\big]\big) \quad (21)$$

where $\mathbb{E}_{p(\cdot)}$ means the expectation of random variables in $p(\cdot)$, $p(\mathbf{Z}^t, \mathbf{H}^t)$ is the joint samples, $p(\mathbf{Z}^t)$ and $p(\mathbf{H}^t)$ stand for the marginal ones, and $T_{\Omega_t} = T_{\Omega_t}(\mathbf{Z}^t, \mathbf{H}^t)$ denotes a multilayer perceptron with parameter Ω_t. Similarly, the counterpart is

$$\begin{aligned}\hat{\mathcal{I}}\big(\mathbf{Z}^f; \mathbf{H}^f\big) =\, & \sup_{\Omega_f}\mathbb{E}_{p(\mathbf{Z}^f,\mathbf{H}^f)}\big[T_{\Omega_f}\big] \\ & - \log\Big(\mathbb{E}_{p(\mathbf{Z}^f)p(\mathbf{H}^f)}\Big[e^{T_{\Omega_f}}\Big]\Big). \quad (22)\end{aligned}$$

F. Model Training

As described in Section III-E, the objective function of SGIB is finally obtained by

$$\begin{aligned}\mathcal{L}_S =\, & \big(\hat{\mathcal{I}}\big(\mathbf{A}^t; \mathbf{H}^t\big) + \hat{\mathcal{I}}\big(\mathbf{A}^f; \mathbf{H}^f\big)\big) - \beta\hat{\mathcal{I}}\big(\mathbf{H}^t; \mathbf{H}^f\big) \\ & - \gamma\big(\hat{\mathcal{I}}\big(\mathbf{Z}^t; \mathbf{H}^t\big) + \hat{\mathcal{I}}\big(\mathbf{Z}^f; \mathbf{H}^f\big)\big). \quad (23)\end{aligned}$$

Furthermore, since the proposed model aims to solve the node classification tasks, a cross-entropy loss is applied to evaluating the distance between the predictive output $\mathbf{Z}$ and the ground truth $\mathbf{Y}^l$, denoted as

$$\mathcal{L}_C = -\sum_{i=1}^{l}\sum_{j=1}^{c}\mathbf{Y}_{ij}^l \ln\mathbf{O}_{ij}. \quad (24)$$

Therefore, the overall loss function of the proposed method is defined as

$$\mathcal{L} = \mathcal{L}_S + \mathcal{L}_C. \quad (25)$$

We also analyze the computation of LGCN-SGIB. In general, the network of LGCN-SGIB can be divided into the following steps: GE, GSL, and graph convolution. Taking topology metapath as an example, the GE process consumes $\mathcal{O}(\text{nct} + \text{nc}^2)$, and GSL costs $\mathcal{O}(n^2 + n^2 f + nf)$, where f is the number of hidden units. Herein, t denotes the number of iterations. The GCNconv process takes $\mathcal{O}(n^2 d + \text{ndf})$. Both the topology metapath and the feature metapath can run efficiently on GPU hardware parallelly. Assuming that $f \approx c$ and $c \ll n$, the overall time complexity of forward calculations in each iteration is $\mathcal{O}(n^2 d)$.

Furthermore, to better clarify the procedure of the proposed framework, we elaborate on the implementation of the whole network, shown in Algorithm 1.

IV. Experiments

In this section, we compare the proposed LGCN-SGIB with nine methods on eight real-world graph datasets. We first briefly describe the experimental setup and then the semisupervised node classification performance of LGCN-SGIB is presented. Finally, comprehensive experiments including parameter sensitivity, ablation study, and convergence validation are conducted to analyze its effectiveness and superiority.

A. Experimental Setup

1) Datasets: Eight benchmark datasets are utilized to perform semisupervised node classification tasks. These graph datasets are shown in Table I. Detailed descriptions are given as follows.

1) **Flickr** is an image and video hosting dataset that forms links between shared Flickr public images, among which the nodes represent users and the edges represent their relationships. All the nodes are divided into nine classes according to the interests of users.
2) **ACM** is a dataset whose nodes stand for the papers and edges stand for the two papers having the same author. All the papers are divided into three classes: citation networks, paper contents, and other data integration studies.

Algorithm 1 Learnable GCN With SGIB (LGCN-SGIB)

Require: Graph $\mathcal{G}_t(\mathbf{A}^t, \mathbf{X})$, label matrix $\mathbf{Y}^l$, epoch number Γ.
Ensure: Predictive output $\mathbf{Z}$.

```
Initialize A^f = kNN(X) by Eq. (3);
Initialize Â^t = Ã^t, Â^f = Ã^f;
for epoch ∈ {1, ..., Γ} do
  for ∗ ∈ {t, f} do
    if epoch > 1 then
      # Global Exploration.
      C̄^* = kmeans(H^*) by Eq. (9);
      Screen nodes X̂_*, and obtain weak-labels Ŷ^* by Eq. (11);
      if x_i, x_j ∈ X̂_* then
        Ā^*_{i,j} = 1 if satisfying ŷ^*_i = ŷ^*_j;
      end if
      Forward propagation in the model;
      # Graph Structural Learning.
      A^* = Mask(Ā^*) using (12);
      Â^* = Attention(Ã^*, A^*) using Eq. (13);
    end if
    # Model Training.
    H^* = GCNconv_{Θ1}(Â^*, X) by Eq. (4);
    Z^* = GCNconv_{Θ2}(Â^*, H^*) by Eq. (5);
  end for
  Z = Z^t + αZ^f by Eq. (8);
  L = L_S + L_C by Eq. (25);
  Back propagation L to update model weights;
end for
return Predictive output Z.
```

TABLE I
STATISTICAL SUMMARY OF ALL GRAPH 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
Cora	2,708	5,429	1,433	7
Film	7,600	15,009	932	5
Citseer	3,327	4,732	3,703	6
UAI	3,067	28,311	4,973	19
CoraFull	19,793	65,311	8,710	70

3) **BlogCatalog** is a social network from a BlogCatalog website. The nodes are made up of the keywords of the user profiles and the edges denote the authors having the same topic categories. All the nodes are divided into six classes.
4) **Cora** is a citation network dataset, where nodes stand for papers and edges stand for their citations between each other. It consists of seven classes and 5429 edges.
5) **Film** is a film social network dataset containing 7600 nodes and 15 009 edges, which is divided into five classes.
6) **Citeseer** is a research paper citation dataset with nodes representing publications and edges representing citation links, where nodes are divided into six categories.
7) **UAI** is a dataset that has 3067 nodes and 28 311 edges. Each feature dimension is 4973, and the nodes are divided into 19 classes.
8) **CoraFull** is a larger version of the well-known Cora dataset consisting of 19 793 nodes and 65 311 edges. All the nodes are divided into 70 classes.

2) Baselines: To validate the proposed framework, we compare our approach with the following state-of-the-art methods.

1) **GCN** [11] is a graph convolutional network applying to the semisupervised classification tasks that obtain information by aggregating messages from neighbors.
2) **kNN-GCN** is a baseline that employs a feature graph constructed by the kNN algorithm to input the GCN for the classification tasks.
3) **GAT** [48] is a graph attention network that applies an attention network to specifying weights for different nodes in neighbors to learn node representations.
4) **GLCN** [27] is a GCN that integrates both graph learning and GCNconv simultaneously in a unified network architecture to learn a graph representation.
5) **Scatter-GCN** [41] is a semisupervised graph neural network that introduces geometric scattering transforms and residual convolutions to alleviate over-smoothing problems for node classification tasks.
6) **AM-GCN** [40] is a multichannel GCN framework that explores GCNconv operation on both topology and feature spaces for the node classification tasks.
7) **CG3-GCN** [29] is a contrastive GCN-based framework that designs a semisupervised contrastive loss to learn transductive node representation.
8) **SSGC** [49] is a neural network extension of the Markov diffusion kernel that captures the global and local contexts of each node simultaneously for the node classification downstream tasks.
9) **NWR-GAE** [50] employs neighborhood Wasserstein reconstruction to build the entire neighborhood information regarding both proximity and structure for node classification tasks.

3) Parameter Settings: In the experiment, all parameter settings of baselines are suggested by their papers. As for the proposed LGCN-SGIB, we shuffle all datasets by randomly selecting $\{20, 40, 60\}$ labeled nodes per class for training, and 500 and 1000 samples for validation and testing, respectively. The Adam optimizer is applied to the learnable parameters with the learning rate as 1×10^{-2} and weight decay as 5×10^{-4}. In the proposed framework, we employ kNN to initialize the feature adjacency matrix $\mathbf{A}^f$, with k ranging from 3 to 50. Besides, the attention mechanism in the GSL module is utilized by a fully-connected neural network, where the hidden neural number is $\{n, 16, 1\}$. In the attention mechanism, $\text{Tanh}(\cdot)$ is used as activation in the first layer, and $\text{Softmax}(\cdot)$ is used in the last layer. We train two GCNconv modules in the topology and feature meta paths, respectively, where the first GCNconv employs $\text{ReLU}(\cdot)$ as neural activation with the hidden dimension being 16, and the second GCNconv adopts $\text{Softmax}(\cdot)$ as neural activation with the output dimension being c. Two well-known metrics including accuracy

TABLE II

NODE CLASSIFICATION PERFORMANCE (ACC% AND F1%) OF ALL COMPARED METHODS WITH 20/40/60 LABELED PER CLASS ON EIGHT DATASETS, WHERE THE BEST RESULTS ARE HIGHLIGHTED IN RED AND THE SECOND BEST RESULTS ARE HIGHLIGHTED IN BLUE

Datasets	L/C	Metrics	GCN	*k*NN-GCN	GAT	GLCN	Scatter-GCN	AM-GCN	CG3-GCN	SSGC	NWR-GAE	**Ours**
Flickr	20	ACC	54.10	**75.90**	43.10	32.20	48.35	74.28	63.50	53.20	44.12	**80.80**
		F1	52.87	**75.65**	39.94	32.18	45.94	73.83	22.62	53.06	43.86	**80.57**
	40	ACC	64.00	77.10	47.30	40.60	59.50	**78.92**	65.30	64.40	46.23	**82.60**
		F1	64.16	77.26	45.38	40.23	57.98	**78.57**	23.03	65.67	46.01	**81.90**
	60	ACC	68.20	78.10	51.36	46.90	64.51	**81.06**	66.30	67.40	47.25	**84.00**
		F1	68.34	77.85	49.40	46.87	63.47	**80.76**	23.33	68.73	46.98	**83.45**
ACM	20	ACC	88.10	86.50	84.75	85.90	89.33	**90.56**	90.10	85.20	61.60	**91.60**
		F1	85.03	86.47	60.54	85.84	89.41	**90.64**	47.15	85.02	60.73	**91.46**
	40	ACC	89.80	87.80	89.08	86.20	90.60	90.60	**91.30**	89.40	61.34	**93.50**
		F1	89.89	87.80	63.14	86.16	**90.66**	90.65	46.58	89.44	60.85	**93.47**
	60	ACC	90.50	88.80	89.71	87.10	90.69	90.26	**91.70**	90.40	61.36	**93.50**
		F1	90.55	88.77	67.32	87.10	**90.75**	90.31	46.71	90.41	60.75	**93.54**
BlogCatalog	20	ACC	85.70	81.30	60.54	61.90	70.44	82.58	82.30	**88.30**	50.79	**91.10**
		F1	85.03	80.93	59.26	58.20	69.03	81.94	31.78	**88.05**	50.49	**91.06**
	40	ACC	88.50	83.40	63.14	66.40	75.18	85.38	85.30	**88.60**	50.77	**91.80**
		F1	88.01	83.24	62.35	66.21	77.22	85.09	31.16	**88.22**	50.63	**91.42**
	60	ACC	89.30	83.50	67.32	69.10	79.23	86.56	83.70	**89.50**	50.76	**92.30**
		F1	88.77	83.29	66.75	65.20	78.59	86.30	31.76	**89.12**	50.75	**92.18**
Cora	20	ACC	**79.10**	69.30	79.10	77.10	78.74	71.57	72.90	76.00	64.40	**82.09**
		F1	77.70	65.98	77.30	76.85	**77.80**	70.99	51.41	73.49	51.68	**81.99**
	40	ACC	79.80	70.10	80.60	78.30	**81.99**	74.62	63.40	79.80	74.70	**82.29**
		F1	77.14	66.30	79.37	77.32	**81.17**	74.18	46.23	77.76	68.41	**82.29**
	60	ACC	81.60	70.40	81.10	78.89	**82.09**	77.52	62.20	81.80	77.00	**82.29**
		F1	77.63	65.43	78.42	78.01	**82.40**	76.98	46.71	77.73	66.52	**82.49**
Film	20	ACC	26.10	**27.10**	19.30	21.70	24.29	26.34	22.00	21.70	14.21	**29.60**
		F1	**26.29**	23.09	19.18	20.59	23.06	17.95	20.33	21.97	14.01	**28.77**
	40	ACC	27.40	**27.10**	21.00	25.80	25.40	26.70	19.90	23.60	14.66	**30.50**
		F1	26.06	**26.98**	20.66	24.69	24.45	20.95	21.24	23.30	14.21	**29.87**
	60	ACC	26.20	**29.30**	22.18	26.40	25.63	27.54	22.19	25.70	14.81	**31.20**
		F1	20.80	24.59	21.90	**25.90**	24.57	21.39	21.08	25.42	14.52	**31.12**
Citeseer	20	ACC	70.30	66.70	66.90	66.70	68.42	**71.22**	69.00	64.70	53.40	**72.20**
		F1	**67.50**	63.31	61.95	63.05	66.07	66.26	44.02	62.81	41.65	**68.90**
	40	ACC	71.61	70.30	68.80	66.40	70.26	**73.40**	72.00	61.90	53.10	**75.30**
		F1	**68.03**	64.68	64.83	63.26	66.86	67.84	44.47	42.59	43.03	**68.70**
	60	ACC	72.86	70.60	68.90	67.80	71.25	**74.20**	73.00	62.30	56.20	**76.30**
		F1	**69.01**	66.62	65.83	63.89	68.11	68.06	44.83	41.15	43.72	**69.54**
UAI	20	ACC	49.20	35.80	58.60	36.10	35.65	**64.30**	57.00	56.10	37.08	**67.00**
		F1	39.32	29.93	40.52	27.20	37.41	**47.80**	47.01	43.47	30.39	**49.58**
	40	ACC	44.20	48.40	60.60	40.50	41.47	**65.90**	63.40	61.90	37.14	**68.00**
		F1	37.83	43.06	40.93	33.36	45.63	**47.20**	46.23	42.59	30.43	**49.92**
	60	ACC	52.60	64.70	62.60	48.30	42.86	**67.80**	62.20	62.30	37.18	**69.30**
		F1	39.11	46.86	42.60	36.90	44.68	**48.10**	46.71	41.15	30.31	**49.95**
CoraFull	20	ACC	**57.01**	45.76	57.44	39.90	54.67	54.76	26.00	56.50	53.69	**57.90**
		F1	**57.10**	45.14	56.68	33.10	46.82	51.05	11.63	35.62	50.21	**57.63**
	40	ACC	**60.52**	45.71	58.39	48.00	59.74	60.10	27.90	60.50	56.89	**60.63**
		F1	**58.49**	44.62	56.89	42.61	48.84	52.12	12.48	33.19	55.20	**59.12**
	60	ACC	**64.17**	45.79	62.19	48.30	64.03	63.75	18.91	63.00	59.67	**64.38**
		F1	**60.12**	43.56	58.62	42.98	50.09	52.32	22.19	26.95	58.12	**60.25**

(ACC) and F1-score (F1) are employed for the performance evaluation. The experiments are conducted under the scenario of 20 labeled samples per class as the training set, without otherwise specified.

B. Semisupervised Node Classification

1) Performance Comparison: We conduct substantial experiments comparing LGCN-SGIB with the selected nine approaches in semisupervised node classification tasks. The performance of different methods with varying labeled training samples is reported in Table II. The experimental results show that LGCN-SGIB performs remarkably on all datasets, while the compared methods take advantage of different datasets, respectively. This observation demonstrates that the proposed framework achieves optimal classification performance, indicating the superior capability of denoising and GSL. Compared with the GCN, *k*NN-GCN, and AM-GCN, the proposed method is superior on all datasets. This may be attributed

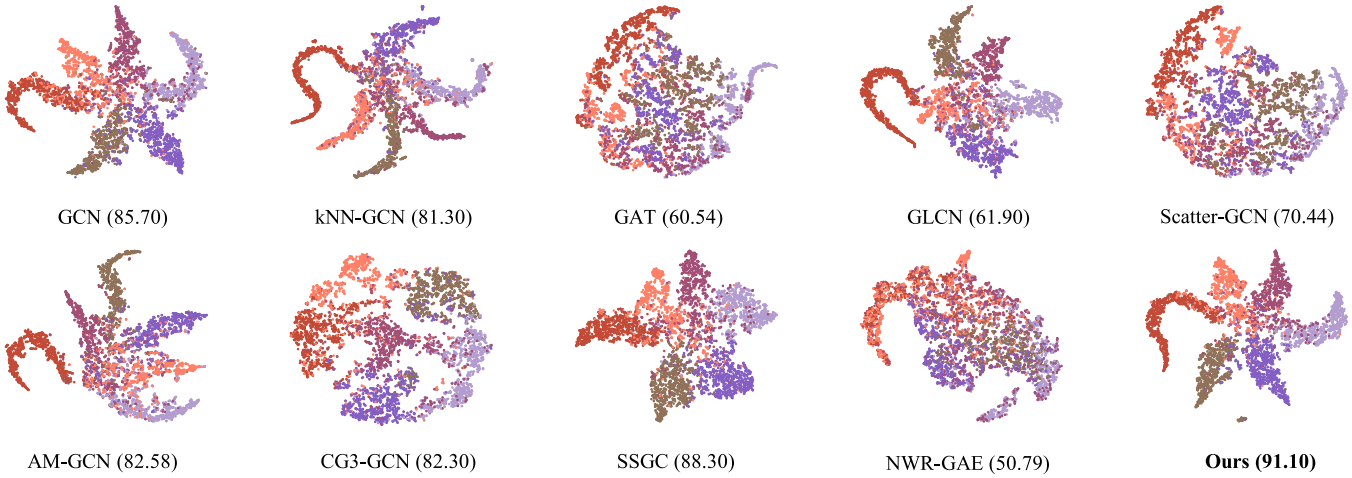

Fig. 3. T-SNE visualization of node classification results of compared methods on BlogCatalog.

to that the compared baselines suffer from the interference of noises in data samples during model training, resulting in undesired performance. Instead, LGCN-SGIB can flexibly refine discriminative graphs, which benefit the downstream tasks through backpropagation and make the networks more robust to noises by learning minimal sufficient representations and filtering useless information. Furthermore, it is noted that GLCN also exhibits node classification performance. Nevertheless, the proposed framework still significantly surpasses GLCN by an average of 14.51%. The reason may be that GLCN takes noisy samples into account when adaptively constructing neighborhood structures, leading to insufficient feature propagation in incorrect topology patterns. While LGCN-SGIB gains stronger graph representations under the guidance of the GSL module and the SGIB principle. For the rest GCN-based methods, LGCN-SGIB also behaves the best, indicating that the proposed framework reasonably obtains more useful sufficient information from the original graphs and learns more valid node representations through the supervision of the GE module, GSL module, and SGIB principle. Overall, LGCN-SGIB can remarkably achieve the best accuracy under different labeled ratios. The results validate that our method can steadily improve performance regardless of whether the information in the original structure is useful or redundant.

2) Visualization of Classification: To intuitively show the classification performance, Fig. 3 presents scatter diagrams of all algorithms on the BlogCatalog dataset with 20 labeled samples per class. From the figures, we can observe that most compared baselines generally succeed in learning separable node features, while some methods mix the nodes belonging to different classes. GCN, SSGC, and the proposed LGCN-SGIB obtain higher classification accuracy and have a stronger ability to assign more accurate class labels. LGCN-SGIB performs even better with closer intraclass correlations and farther extraclass correlations, which may be attributed to the SGIB that alleviates the interference of noises in the original data and strengthens the consistency of both the same meta-channel and different meta-channels. These observations verify the superiority of node representation learning ability for the proposed LGCN-SGIB.

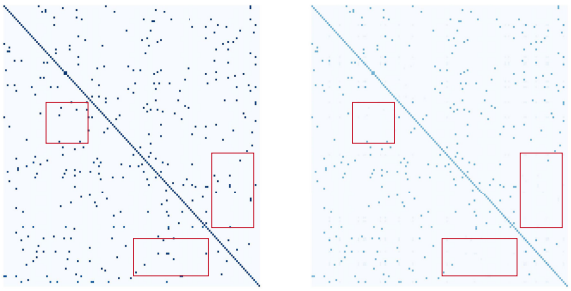

Fig. 4. Visualization of the adjacency matrix $\mathbf{A}^f$ (left) and the adjacency matrix $\hat{\mathbf{A}}^f$ (right) learned by LGCN-SGIB in feature metapath on the Citeseer dataset, where darker colors are the higher weighted values of matrices and red boxes highlight disappeared edges.

3) Refined Adjacency Matrix: Fig. 4 presents the visualization of the partial initial adjacency matrix and adjacency matrix learned by LGCN-SGIB in the feature metapath. It can be seen that some connections in the learned adjacency matrix diminish or disappear under the guidance of the SGIB principle, thereby making graphs more sparser and robust. Furthermore, compared with the initial structure, the adjacency matrix refined by the SGIB principle is relatively pure, which is beneficial for node embedding learning. The decent performance also favors the superiority of LGCN-SGIB.

C. Module Analysis

1) Convergence Validation: Fig. 5 illustrates the curves of LGCN-SGIB in terms of train loss, valid loss, and valid accuracy under selecting 20 labeled samples per class for the training set. We have the following observations. With the step size of 0.01, it is evident that both train loss and valid loss decrease and eventually converge within 100 iterations on most datasets. Besides, the accuracy of the validation set generally grows and tends to be convergent. Nevertheless, the valid loss on the Film dataset slightly rises with an increasing number of training iterations, which may be reduced to the instability of the primary model. In this article, we select the optimal model with the lowest valid loss value to perform the node classification prediction on the test set.

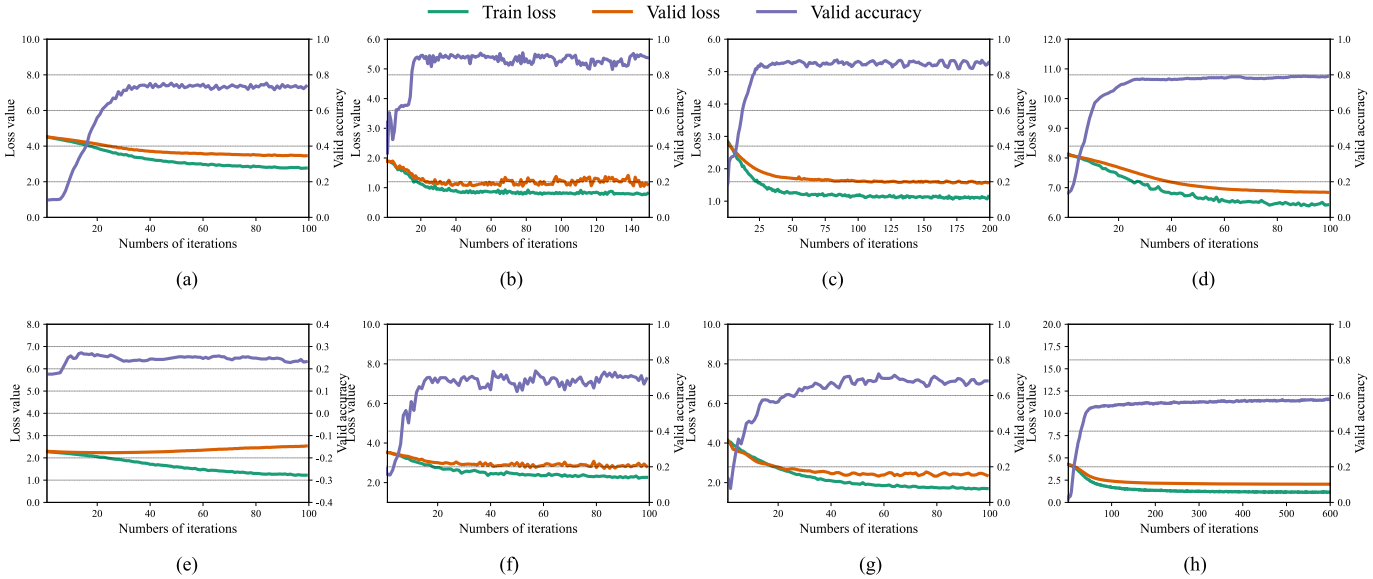

Fig. 5. Training loss (green), valid loss (orange), and valid accuracy (purple) on all datasets. (a) Flickr. (b) ACM. (c) BlogCatalog. (d) Cora. (e) Film. (f) Citeseer. (g) UAI. (h) CoraFull.

Predict Labels

Ground-truth	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19	SUM
C1	0.522	0.030	0	0	0.045	0	0	0.060	0.104	0.030	0.075	0.104	0	0	0	0.015	0	0.015	0	67
C2	0.074	0.526	0	0.011	0.040	0.063	0.006	0.034	0.046	0	0.097	0.034	0	0.023	0	0	0.006	0.040	0	175
C3	0	0	0.855	0	0	0.016	0.032	0	0.081	0	0	0	0	0	0	0	0	0.016	0	62
C4	0	0	0	0.900	0.050	0	0	0.050	0	0	0	0	0	0	0	0	0	0	0	20
C5	0	0	0	0	1.000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
C6	0.074	0	0	0.007	0.013	0.617	0.013	0.040	0.094	0.020	0.054	0.013	0.020	0	0.007	0.027	0	0	0	149
C7	0.056	0	0	0	0	0.056	0.741	0	0.037	0	0	0.037	0	0.074	0	0	0	0	0	54
C8	0.032	0.032	0	0	0.129	0	0	0.581	0.129	0.032	0	0.065	0	0	0	0	0	0	0	31
C9	0.140	0	0.033	0	0.017	0.025	0	0.107	0.554	0	0.017	0.041	0.025	0.017	0	0.025	0	0	0	121
C10	0	0	0	0	0.100	0	0	0	0	0.800	0	0	0.100	0	0	0	0	0	0	10
C11	0.126	0.010	0	0	0	0	0	0.078	0.078	0	0.631	0	0	0	0	0	0.068	0.010	0	103
C12	0	0	0	0	0	0	0	0	0	0	0	1.000	0	0	0	0	0	0	0	20
C13	0	0	0	0	0	0.058	0.019	0.019	0.019	0	0	0	0.885	0	0	0	0	0	0	52
C14	0	0	0	0	0	0	0	0	0	0	0	0	0	1.000	0	0	0	0	0	2
C15	0	0	0	0	0	0	0	0	0	0	0	0.250	0	0	0.750	0	0	0	0	4
C16	0	0	0	0	0	0	0	0	0.200	0	0	0	0	0	0	0.800	0	0	0	5
C17	0	0	0	0	0	0	0	0	0.077	0	0	0	0	0.077	0	0	0.846	0	0	13
C18	0.111	0.012	0	0	0	0.025	0	0.012	0	0	0.012	0	0	0	0	0	0	0.827	0	81
C19	0	0	0	0.034	0	0	0	0	0	0	0	0	0	0	0	0	0	0.034	0.931	29

Fig. 6. Row-normalized confusion matrix from the test set of the UAI dataset, where values represent the probabilities of the corresponding categories.

2) Error Analysis: To investigate the reason for the misclassification across all categories, we conducted additional experiments on the UAI dataset by analyzing the difference between the ground truth and the prediction label distributions. As shown in Fig. 6, we visualize the true class and predicted class by a confusion matrix and enumerate the number of nodes in each category. The top five misclassification categories {C1, C2, C6, C8, C9} are marked by dark blue. From the figure, we can observe that classes with more nodes have a higher classification error probability. The rationale behind this phenomenon is that classes with more nodes also possess more edges connecting them to other nodes. Consequently, the likelihood of anomalous edge interference increases, leading to undesired feature propagation and suboptimal performance. The observation indicates that it is challenging to predict classes with higher edge densities compared to other subtypes.

3) Parameter Sensitivity: To illustrate the performance variations of the proposed method under different settings, parameter sensitivity analysis is conducted with respect to various hyperparameters β and γ, as shown in Figs. 7 and 8. In our model, β and γ are used to balance the weights of two terms in the objective function of the SGIB principle. We fix one of the hyperparameters at the best value and modify the other to observe the influence on the test set. From the figures, we can conclude that the selection of these two hyperparameters has a remarkable influence on the classification performance. Specifically, the accuracy and F1-score are at lower levels when the hyperparameter β is too large, and they tend to be stable when β is less than 1. Based on this, the suggested value for β ranges from 1×10^{-5} to 1×10^{-1}. In addition, the classification performance of the model is relatively stable with different values of parameter γ.

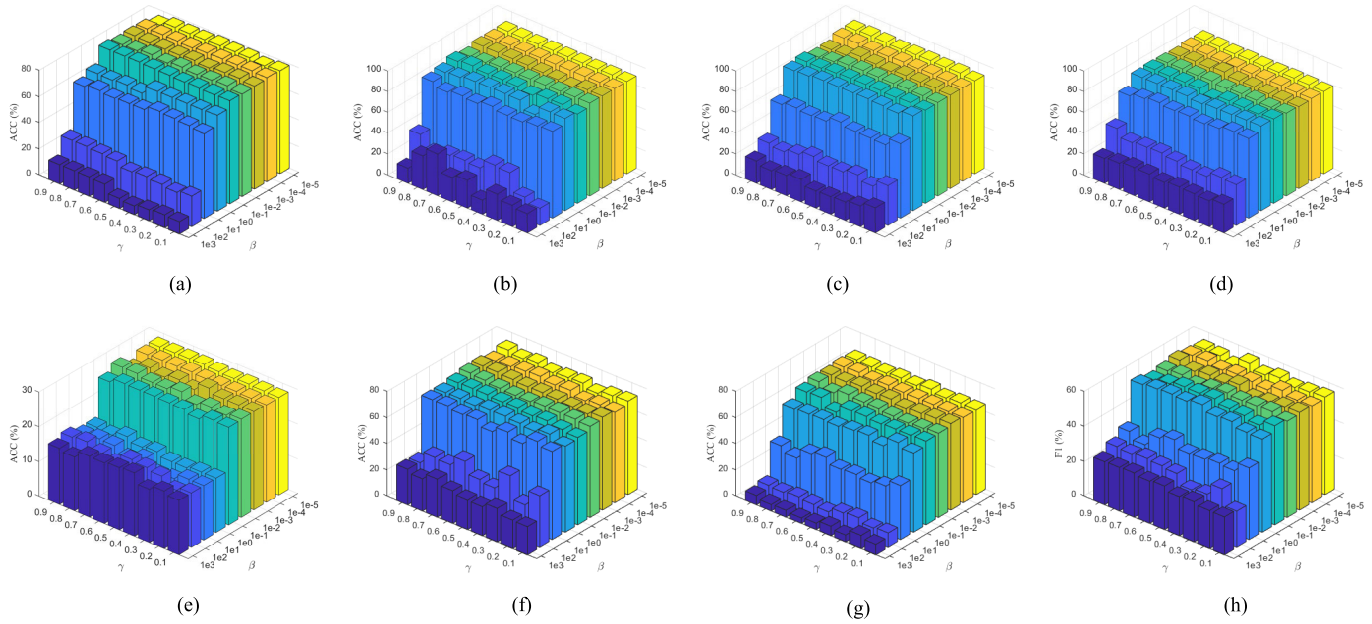

Fig. 7. Parameter sensitivity (ACC%) of the proposed method with respect to hyperparameters β and γ on all datasets. (a) Flickr. (b) ACM. (c) BlogCatalog. (d) Cora. (e) Film. (f) Citeseer. (g) UAI. (h) CoraFull.

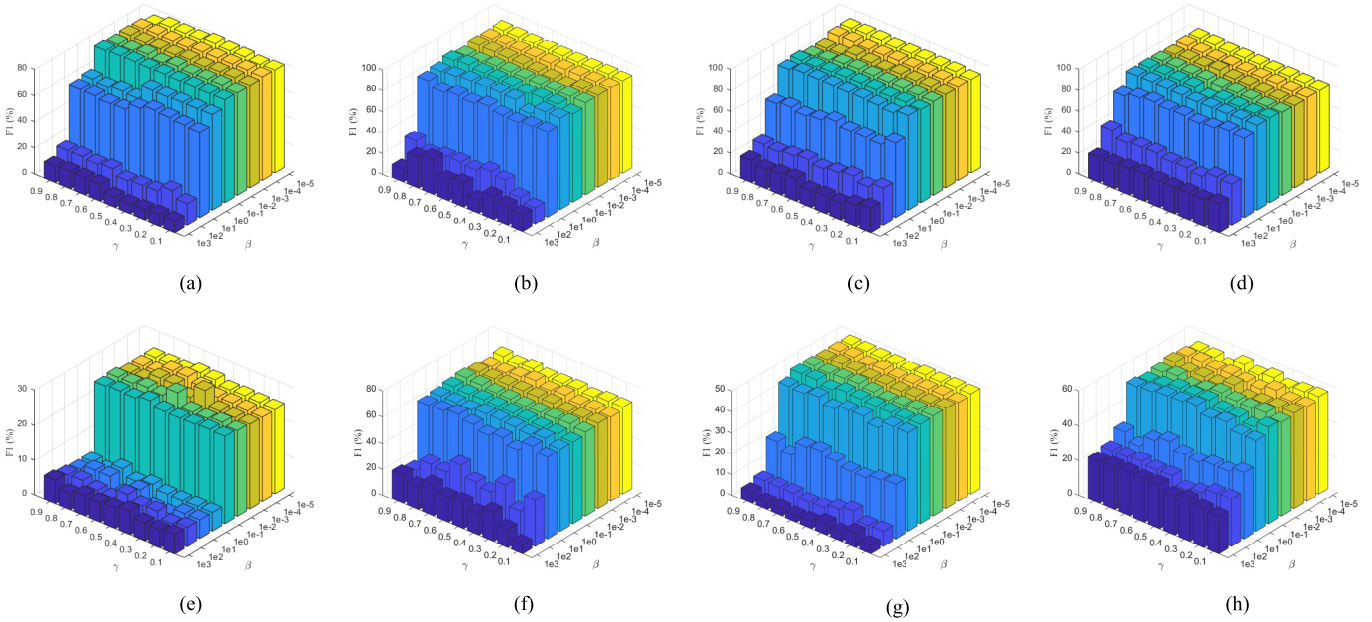

Fig. 8. Parameter sensitivity (F1%) of the proposed method with respect to hyperparameters β and γ on all datasets. (a) Flickr. (b) ACM. (c) BlogCatalog. (d) Cora. (e) Film. (f) Citeseer. (g) UAI. (h) CoraFull.

Generally, smaller or larger values of γ lead to poor performance, and satisfactory performance is achieved with γ in a moderate range. Therefore, the suggested value of γ ranges in [0.5, 0.7].

4) Ablation Study: To intuitively validate the contribution of the proposed modules, we test the classification accuracy and F1-score of LGCN-SGIB with its variants on all datasets, shown in Table III. Our model can be divided into three components: the GE module, the GSL module, and the SGIB module. We fist utilize the GCN as a baseline and then gradually stack the proposed modules to analyze the effectiveness. It is worth noting that the proposed framework employs SGIB loss and cross-entropy loss as the loss function, and the rest variants employ cross-entropy loss as an objective function. From the table, we can observe that the accuracy goes up when the modules are stacked one by one. In addition, it is clear that using one of the three individually leads to a smaller performance improvement, while it achieves optimal accuracy when integrating all of them. This validates that LGCN-SGIB obtains more desired graph structures from local and global perspectives for promoting effective feature propagation. Furthermore, compared with LGCN-SGIB, these variants without the guidance of the SGIB principle neglect to minimize the redundant information from the initial structures, which means

TABLE III
ABLATION STUDY (ACC% AND F1%) OF THE PROPOSED METHOD ON ALL DATASETS

Datasets	GE	GSL	SGIB	ACC	F1	Datasets	GE	GSL	SGIB	ACC	F1
Flickr	✗	✗	✗	54.10	52.87	ACM	✗	✗	✗	88.10	85.03
	✓	✗	✗	79.00	78.85		✓	✗	✗	89.10	89.00
	✓	✓	✗	79.30	79.20		✓	✓	✗	90.10	89.77
	✓	✓	✓	**80.80**	**80.57**		✓	✓	✓	**91.60**	**91.46**
BlogCatalog	✗	✗	✗	85.70	85.03	Cora	✗	✗	✗	79.10	77.70
	✓	✗	✗	89.20	88.89		✓	✗	✗	79.28	79.35
	✓	✓	✗	91.10	90.84		✓	✓	✗	79.68	79.82
	✓	✓	✓	**91.10**	**91.06**		✓	✓	✓	**82.09**	**81.99**
Film	✗	✗	✗	26.10	26.29	Citeseer	✗	✗	✗	70.30	67.50
	✓	✗	✗	28.10	26.20		✓	✗	✗	69.10	64.16
	✓	✓	✗	28.50	28.20		✓	✓	✗	70.40	65.50
	✓	✓	✓	**29.60**	**28.77**		✓	✓	✓	**72.20**	**68.90**
UAI	✗	✗	✗	49.20	39.32	CoraFull	✗	✗	✗	57.01	57.10
	✓	✗	✗	62.30	43.52		✓	✗	✗	57.16	57.10
	✓	✓	✗	63.00	46.01		✓	✓	✗	57.16	57.10
	✓	✓	✓	**67.00**	**49.58**		✓	✓	✓	**57.90**	**57.63**

that the learned representation may be severely affected by the noises from the original graph. On the contrary, the utilization of SGIB aims to minimize the mutual information between node representations and the initial structure, which alleviates the interference of perturbations from the original data and improves the consistency of meta-channels.

V. CONCLUSION

In this article, we proposed a learnable GCN framework named LGCN-SGIB to cope with the node classification problem, which built dual-meta-channels to dynamically explore potential graph information from both topology and feature spaces. We first employed the GE module as a supervised signal to fully extract connective relationships from hidden spaces. Then the GSL module was employed to adaptively learn optimal graph structures deriving from global and local information. Finally, the SGIB principle was conducted to evaluate the mutual information of the same and different meta-channels to enhance the consistency of valid information while mitigating the interference of noises, simultaneously. The mutual information between graph structures and predictive outputs evaluated the number of learned graph structures, promoting the interpretability of the proposed framework. Substantial experimental results demonstrated the effectiveness and superiority compared with state-of-the-art models.

In this work, we only focus on denoising the topology and feature metapaths in tLGCN-SGIBhe centralized scenarios. However, in real-world applications, it is important to protect data privacy in distributed environments. Since SGIB can extract minimal sufficient information, it would be urgent to investigate whether the sensitive information implied in data could be removed for fairness. Therefore, in future work, we will explore a theoretical analysis of the privacy guarantee provided by this framework.

APPENDIX

A. Proof of Lemma 1

We restate Lemma 1, which is in the fourth paragraph of Part E, Section III.

Lemma 1. Given a latent variable M and two random variables X_1 and X_2, we have

$$\mathcal{I}(X_1; X_2) \geq \mathbb{E}_{q(X_1,X_2|M=1)}\big[\log q(M=1|X_1, X_2)\big] + \mathbb{E}_{q(X_1,X_2|M=0)}\big[\log q(M=0|X_1, X_2)\big] \quad (25)$$

where $M = 1$ denotes that (x_1, x_2) belongs to the same class sampling from the joint distribution and $M = 0$ represents that (x_1, x_2) belongs to different classes from the marginal distribution.

Proof: Since $M = 1$ represents that (x_1, x_2) have the same class that derives from the joint distribution, and, therefore, $q(X_1, X_2|M = 1)$ denotes the joint distribution equaling to $q(X_1, X_2)$. Similarly, $q(X_1, X_2|M = 0)$ equals to $q(X_1)q(X_2)$ sampling from the marginal distribution. Meanwhile, given the known conditions, we have $q(M = 1) = (l/N)$ and $q(M = 0) = (N - l/N)$ $(l \ll N)$. Naturally, the poster probability of $M = 1$ is defined as

$$\begin{aligned}
&\log q(M=1|X_1, X_2) \\
&= \log \frac{q(X_1, X_2|M=1)q(M=1)}{\sum_{j\in[0,1]} q(X_1, X_2|M=j)q(M=j)} \\
&= \log \frac{q(X_1, X_2)}{\frac{N-l}{l}q(X_1)q(X_2) + q(X_1, X_2)} \\
&= -\log\left(\frac{N-l}{l}\frac{q(X_1)q(X_2)}{q(X_1, X_2)} + 1\right) \\
&\leq -\log\left(\frac{N-l}{l}\right) - \log\frac{q(X_1)q(X_2)}{q(X_1, X_2)}
\end{aligned}$$

therefore, ignoring the constant, the mutual information is written as

$$\mathcal{I}(X_1, X_2) \geq \mathbb{E}_{q(X_1,X_2|M=1)} \log q(M = 1|X_1, X_2).$$

To consider the poster probability of $M = 0$, simultaneously, the mutual information $\mathcal{I}(X_1, X_2)$ is further written as

$$\begin{aligned}\mathcal{I}(X_1, X_2) \geq & \mathbb{E}_{q(X_1,X_2|M=1)} \log q(M = 1|X_1, X_2) \\ & + \mathbb{E}_{q(X_1,X_2|M=0)} \log q(M = 0|X_1, X_2).\end{aligned}$$

Since the known condition $\mathbb{E}_{q(M=0|X_1,X_2)} \log q(M = 0|X_1, X_2) \leq 0$, (25) holds, completing the proof. □

B. Derivation of (15)

We restate to validate (15), which is in the second paragraph of Part E, Section III

$$\begin{aligned}\hat{\mathcal{I}}(\mathbf{A}^t; \mathbf{H}^t) \leq & \hat{\mathcal{I}}(\mathbf{A}^t, \hat{\mathbf{A}}^t) \\ = & D_{KL}\big(p(\hat{\mathbf{A}}^t|\mathbf{A}^t) \,||\, r(\mathbf{A}^t)\big). \end{aligned} \tag{15}$$

Proof: Given mutual information $\mathcal{I}(\mathbf{A}^t, \hat{\mathbf{A}}^t)$, we have

$$\begin{aligned}\mathcal{I}(\mathbf{A}^t, \hat{\mathbf{A}}^t) = & \mathbb{E}_{p(\mathbf{A}^t,\hat{\mathbf{A}}^t)}\left[\log \frac{p(\mathbf{A}^t, \hat{\mathbf{A}}^t)}{p(\mathbf{A}^t)p(\hat{\mathbf{A}}^t)}\right] \\ = & \mathbb{E}_{p(\mathbf{A}^t,\hat{\mathbf{A}}^t)}\left[\log \frac{p(\mathbf{A}^t|\hat{\mathbf{A}}^t)}{p(\hat{\mathbf{A}}^t)}\right] \\ = & \mathbb{E}_{p(\hat{\mathbf{A}}^t|\mathbf{A}^t)p(\mathbf{A}^t)}\left[\log \frac{p(\mathbf{A}^t|\hat{\mathbf{A}}^t)}{p(\hat{\mathbf{A}}^t)}\right] \\ = & \mathcal{D}_{KL}\big(p(Y|\mathbf{A}^t)||p(\hat{\mathbf{A}}^t)\big)p(\hat{\mathbf{A}}^t).\end{aligned}$$

Since the nonnegative of KL divergence and the value $p(\hat{\mathbf{A}}^t) \in [0, 1]$, we have

$$\begin{aligned}\mathcal{I}(\mathbf{A}^t, \hat{\mathbf{A}}^t) \leq & \hat{\mathcal{I}}(\mathbf{A}^t, \hat{\mathbf{A}}^t) \\ = & \mathcal{D}_{KL}\big(p(Y|\mathbf{A}^t)||p(\hat{\mathbf{A}}^t)\big)\end{aligned}$$

which completes the proof. □

REFERENCES

[1] Z. Li, H. Liu, Z. Zhang, T. Liu, and N. N. Xiong, "Learning knowledge graph embedding with heterogeneous relation attention networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 8, pp. 3961–3973, Aug. 2022.

[2] Z. Sun et al., "Knowledge graph alignment network with gated multi-hop neighborhood aggregation," in *Proc. 34th AAAI Conf. Artif. Intell.*, 2020, pp. 222–229.

[3] B. C. Molokwu and Z. Kobti, "Social network analysis using RLVECN: Representation learning via knowledge-graph embeddings and convolutional neural-network," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, Jul. 2020, pp. 5198–5199.

[4] Z. Chen, Z. Wu, Z. Lin, S. Wang, C. Plant, and W. Guo, "AGNN: Alternating graph-regularized neural networks to alleviate over-smoothing," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, May 31, 2023, doi: 10.1109/TNNLS.2023.3271623.

[5] I. Spinelli, S. Scardapane, and A. Uncini, "Adaptive propagation graph convolutional network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 10, pp. 4755–4760, Oct. 2021.

[6] Q. Xie, Y. Zhu, J. Huang, P. Du, and J.-Y. Nie, "Graph neural collaborative topic model for citation recommendation," *ACM Trans. Inf. Syst.*, vol. 40, no. 3, pp. 1–30, Jul. 2022.

[7] Z. Chen, Z. Wu, S. Wang, and W. Guo, "Dual low-rank graph autoencoder for semantic and topological networks," in *Proc. 37th AAAI Conf. Artif. Intell.*, 2023, pp. 4191–4198.

[8] Y. Wang et al., "DisenCite: Graph-based disentangled representation learning for context-specific citation generation," in *Proc. 36th AAAI Conf. Artif. Intell.*, 2022, pp. 11449–11458.

[9] J. Park, M. Lee, H. J. Chang, K. Lee, and J. Y. Choi, "Symmetric graph convolutional autoencoder for unsupervised graph representation learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6518–6527.

[10] L. Wu et al., "Beyond homophily and homogeneity assumption: Relation-based frequency adaptive graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jan. 6, 2023, doi: 10.1109/TNNLS.2022.3230417.

[11] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. 5th Int. Conf. Learn. Represent.*, 2017, pp. 1–14.

[12] Z. Chen, L. Fu, J. Yao, W. Guo, C. Plant, and S. Wang, "Learnable graph convolutional network and feature fusion for multi-view learning," *Inf. Fusion*, vol. 95, pp. 109–119, Jul. 2023.

[13] Y. Bi, A. Chadha, A. Abbas, E. Bourtsoulatze, and Y. Andreopoulos, "Graph-based spatio-temporal feature learning for neuromorphic vision sensing," *IEEE Trans. Image Process.*, vol. 29, pp. 9084–9098, 2020.

[14] X. Liu, Z. Ji, Y. Pang, J. Han, and X. Li, "DGIG-Net: Dynamic graph-in-graph networks for few-shot human–object interaction," *IEEE Trans. Cybern.*, vol. 52, no. 8, pp. 7852–7864, Aug. 2022.

[15] F. Feng, X. He, H. Zhang, and T.-S. Chua, "Cross-GCN: Enhancing graph convolutional network with k-order feature interactions," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 1, pp. 225–236, Jan. 2023.

[16] X. Yang, Y. Lyu, T. Tian, Y. Liu, Y. Liu, and X. Zhang, "Rumor detection on social media with graph structured adversarial learning," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, Jul. 2020, pp. 1417–1423.

[17] L. Pasa, N. Navarin, W. Erb, and A. Sperduti, "Empowering simple graph convolutional networks," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jan. 2, 2023, doi: 10.1109/TNNLS.2022.3232291.

[18] J. Cheng, Q. Wang, Z. Tao, D. Xie, and Q. Gao, "Multi-view attribute graph convolution networks for clustering," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, Jul. 2020, pp. 2973–2979.

[19] S. Xiao, S. Du, Z. Chen, Y. Zhang, and S. Wang, "Dual fusion-propagation graph neural network for multi-view clustering," *IEEE Trans. Multimedia*, early access, Feb. 23, 2023, doi: 10.1109/TMM.2023.3248173.

[20] S. Xiao, S. Wang, and W. Guo, "SGAE: Stacked graph autoencoder for deep clustering," *IEEE Trans. Big Data*, vol. 9, no. 1, pp. 254–266, Feb. 2023.

[21] L. Yang, Y. Guo, J. Gu, D. Jin, B. Yang, and X. Cao, "Probabilistic graph convolutional network via topology-constrained latent space model," *IEEE Trans. Cybern.*, vol. 52, no. 4, pp. 2123–2136, Apr. 2022.

[22] L. Yao, C. Mao, and Y. Luo, "Graph convolutional networks for text classification," in *Proc. 33rd AAAI Conf. Artif. Intell.*, 2019, pp. 7370–7377.

[23] A. Pareja et al., "EvolveGCN: Evolving graph convolutional networks for dynamic graphs," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2020, vol. 34, no. 4, pp. 5363–5370.

[24] S. Wang, Z. Chen, S. Du, and Z. Lin, "Learning deep sparse regularizers with applications to multi-view clustering and semi-supervised classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 5042–5055, Sep. 2022.

[25] T. Wu, H. Ren, P. Li, and J. Leskovec, "Graph information bottleneck," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 20437–20448.

[26] L. Peng et al., "Reverse graph learning for graph neural network," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Apr. 5, 2022, doi: 10.1109/TNNLS.2022.3161030.

[27] B. Jiang, Z. Zhang, D. Lin, J. Tang, and B. Luo, "Semi-supervised learning with graph learning-convolutional networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11305–11312.

[28] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, "Deep variational information bottleneck," in *Proc. 5th Int. Conf. Learn. Represent.*, 2017, pp. 1–19.

[29] S. Wan, S. Pan, J. Yang, and C. Gong, "Contrastive and generative graph convolutional networks for graph-based semi-supervised learning," in *Proc. 35th AAAI Conf. Artif. Intell.*, 2021, pp. 10049–10057.

[30] Q. Sun et al., "Graph structure learning with variational information bottleneck," in *Proc. 36th AAAI Conf. Artif. Intell.*, 2022, pp. 4165–4174.

[31] K. Chen, L. Yao, D. Zhang, X. Wang, X. Chang, and F. Nie, "A semisupervised recurrent convolutional attention model for human activity recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 5, pp. 1747–1756, May 2020.

[32] W. Guan, F. Jiao, X. Song, H. Wen, C.-H. Yeh, and X. Chang, "Personalized fashion compatibility modeling via metapath-guided heterogeneous graph learning," in *Proc. 45th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2022, pp. 482–491.

[33] M. Luo, X. Chang, L. Nie, Y. Yang, A. G. Hauptmann, and Q. Zheng, "An adaptive semisupervised feature analysis for video semantic recognition," *IEEE Trans. Cybern.*, vol. 48, no. 2, pp. 648–660, Feb. 2018.

[34] W. Guan, X. Song, H. Zhang, M. Liu, C.-H. Yeh, and X. Chang, "Bi-directional heterogeneous graph hashing towards efficient outfit recommendation," in *Proc. 30th ACM Int. Conf. Multimedia*, Oct. 2022, pp. 268–276.

[35] Z. Wu, L. Shu, Z. Xu, Y. Chang, C. Chen, and Z. Zheng, "Robust tensor graph convolutional networks via T-SVD based graph augmentation," in *Proc. 28th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2022, pp. 2090–2099.

[36] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1025–1035.

[37] M. Niepert, M. Ahmed, and K. Kutzkov, "Learning convolutional neural networks for graphs," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 2014–2023.

[38] L. Zhong, J. Yang, Z. Chen, and S. Wang, "Contrastive graph convolutional networks with generative adjacency matrix," *IEEE Trans. Signal Process.*, vol. 71, pp. 772–785, 2023, doi: 10.1109/TSP.2023.3254888.

[39] K. Xu, H. Huang, P. Deng, and Y. Li, "Deep feature aggregation framework driven by graph convolutional network for scene classification in remote sensing," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 10, pp. 5751–5765, Oct. 2022.

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

[41] Y. Min, F. Wenkel, and G. Wolf, "Scattering GCN: Overcoming oversmoothness in graph convolutional networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 14498–14508.

[42] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger, "Simplifying graph convolutional networks," in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 6861–6871.

[43] K. Sun, Z. Lin, and Z. Zhu, "Multi-stage self-supervised learning for graph convolutional networks on graphs with few labeled nodes," in *Proc. 34th AAAI Conf. Artif. Intell.*, 2020, pp. 5892–5899.

[44] C. Yan, X. Chang, M. Luo, H. Liu, X. Zhang, and Q. Zheng, "Semantics-guided contrastive network for zero-shot object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Jan. 4, 2022, doi: 10.1109/TPAMI.2021.3140070.

[45] J. Yu, T. Xu, Y. Rong, Y. Bian, J. Huang, and R. He, "Graph information bottleneck for subgraph recognition," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1–13.

[46] L. Yang et al., "Heterogeneous graph information bottleneck," in *Proc. 30th Int. Joint Conf. Artif. Intell.*, Aug. 2021, pp. 1638–1645.

[47] X. Nguyen, M. J. Wainwright, and M. I. Jordan, "Estimating divergence functionals and the likelihood ratio by convex risk minimization," *IEEE Trans. Inf. Theory*, vol. 56, no. 11, pp. 5847–5861, Nov. 2010.

[48] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lió, and Y. Bengio, "Graph attention networks," in *Proc. 6th Int. Conf. Learn. Represent.*, 2018, pp. 1–12.

[49] H. Zhu and P. Koniusz, "Simple spectral graph convolution," in *Proc. 9th Int. Conf. Learn. Represent.*, 2021, pp. 1–15.

[50] M. Tang, P. Li, and C. Yang, "Graph auto-encoder via neighborhood Wasserstein reconstruction," in *Proc. 10th Int. Conf. Learn. Represent.*, 2022, pp. 1–17.

Luying Zhong received the B.S. degree from the College of Computer and Data Science, Fuzhou University, Fuzhou, China, in 2023, where she is currently pursuing the Ph.D. degree.

Her current research interests include graph learning, edge computing, and federated learning.

Zhaoliang Chen received the 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.

He is currently a Visiting Researcher with the Faculty of Computer Science, University of Vienna, Vienna, Austria. His current research interests include feature selection, matrix factorization, graph neural networks, and deep learning.

Zhihao Wu received the B.S. degree from the College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China, in 2021, where he is currently pursuing the M.S. degree with the College of Computer and Data Science, Fuzhou University.

He is currently a Visiting Researcher with the Shenzhen Research Institute of Big Data, The Chinese University of Hong Kong (Shenzhen), Shenzhen, China. His current research interests include machine learning, multiview learning, and graph neural networks.

Shide Du received the M.S. degree from the College of Computer and Data Science, Fuzhou University, Fuzhou, China, in 2022, where he is currently pursuing the Ph.D. degree.

His current research interests include machine learning, differentiable programming, and deep learning.

Zheyi Chen (Associate Member, IEEE) received the M.Sc. degree in computer science and technology from Tsinghua University, Beijing, China, in 2017, and the Ph.D. degree in computer science from the University of Exeter, Exeter, U.K., in 2021.

He is currently a Professor and a Qishan Scholar with the College of Computer and Data Science, Fuzhou University, Fuzhou, China. He has published over 20 research papers in reputable international journals and conferences such as IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE *Communications Magazine*, IEEE TRANSACTIONS ON CLOUD COMPUTING, IEEE *IoT Journal*, and IEEE ICC. His research interests include cloud-edge computing, resource optimization, deep learning, and reinforcement learning.

Shiping Wang (Senior Member, IEEE) 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, Singapore, from August 2015 to August 2016. He is currently a Full Professor and a 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.