Structure-Preserving Graph Representation Learning

Ruiyi Fang[1], Liangjian Wen[2], Zhao Kang[3], Jianzhuang Liu[2]
[1]School of Information and Software Engineering, University of Electronic Science and Technology of China
[2]The Noah's Ark Lab, Huawei Technologies Company Limited
[3]School of Computer Science and Engineering, University of Electronic Science and Technology of China
fangruiyi84@gmail.com, zkang@uestc.edu.cn, {wenliangjian1, liu.jianzhuang}@huawei.com

***Abstract*—Though graph representation learning (GRL) has made significant progress, it is still a challenge to extract and embed the rich topological structure and feature information in an adequate way. Most existing methods focus on local structure and fail to fully incorporate the global topological structure. To this end, we propose a novel Structure-Preserving Graph Representation Learning (SPGRL) method, to fully capture the structure information of graphs. Specifically, to reduce the uncertainty and misinformation of the original graph, we construct a feature graph as a complementary view via k-Nearest Neighbor method. The feature graph can be used to contrast at node-level to capture the local relation. Besides, we retain the global topological structure information by maximizing the mutual information (MI) of the whole graph and feature embeddings, which is theoretically reduced to exchanging the feature embeddings of the feature and the original graphs to reconstruct themselves. Extensive experiments show that our method has quite superior performance on semi-supervised node classification task and excellent robustness under noise perturbation on graph structure or node features. The source code is available at https://github.com/uestc-lese/SPGRL.**

***Index Terms*—Mutual information, contrastive learning, semi-supervised classification, graph convolutional network**

I. Introduction

Ubiquitous graph or network data expressed in the form of node connections and features raise a new challenge for traditional machine learning techniques to discover knowledge [1]. Graph convolutional network (GCN) has proved to be a powerful tool to handle graph-structured data in a variety of domains, such as social network, chemistry, biology, traffic prediction, text classification, and knowledge graph. Most GCN-based methods learn a low-dimensional and dense representation by reconstructing the feature or graph in the autoencoder framework [2]. How to fully inherit the rich information from topological structure and node attribute is crucial to the success of GCN [3].

Basically, GCN processes graph by means of aggregating features from neighborhood nodes. In essence, it performs as low-pass filtering on feature vectors of nodes and graph structure only provides a way to denoise the data [4]. Some works have theoretically analyzed the weaknesses of GCN in feature information fusion [5]. Unlike some other deep neural networks, stacking multiple layers leads to over-smoothing, which seriously degrades the feature discriminability and deteriorates the performance of downstream tasks [6].

§ Ruiyi Fang and Liangjian Wen have equal contributions.
* Zhao Kang is the corresponding author.

In order to better fuse the feature information, graph attention network (GAT) [7] has been proposed, which can assign an adaptive weight to each edge of the graph. Later, Wang *et al.* [5] propose adaptive multi-channel GCN (AMGCN), which better fuses the topological structure and feature information through the attention mechanism. However, the attention-based approach needs to calculate the weight of each edge, which consumes much computation time and memory for large graphs.

Recently, contrastive learning, as a burgeoning unsupervised learning mechanism, has achieved superior performance in various tasks [8]. It learns effective representations by contrasting positive samples against negative samples through the design of pretext tasks including the design of data augmentation schemes and object functions. Some representative works are GRACE [9], SLAPS [10], GCA [11]. These methods mainly explore the local relation without preserving structural information. Recently, maximizing mutual information (MI) has been adopted to explore rich information from topological structure and node features. Deep Graph InfoMax (DGI) [12] maximizes the MI between the hidden representation and a summary vector. However, its simple averaging readout function damages the distinguish capability between nodes and makes the global-level representation unreliable. These methods largely rely on "augmentation engineering", which requires extensive domain knowledge and even incurs negative effects.

To get rid of above issue, some other methods use two neural networks to learn from each other to boost performance. For example, SCRL [13] performs representation consistency constraint by constructing feature graph and topology graph for cross-prediction, and effectively improves the feature information fusion ability of GCN. Some other data augmentation strategies, such as GEN [14] and PTDNet [15], have also been developed. Graphical Mutual Information (GMI) [16] instead tries to maximize the MI between the target node and its neighbors at node-level, and the proximity topological structure at the edge level. As shown in Fig.1, maximizing MI at multiple levels does not really consider MI at the global level. They mainly align embeddings between the same nodes in different topological structures, rather than using the local-global relationships.

To better explore the global structure, we propose a novel Structure-Preserving Graph Representation Learning (SPGRL) method, which maximizes the MI between topology graph

2374-8486/22/$31.00 ©2022 IEEE
DOI 10.1109/ICDM54844.2022.00109

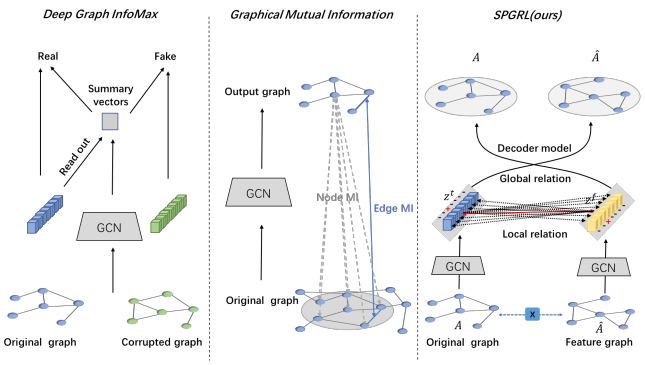

Fig. 1. An overview of DGI (left), GMI (middle) and SPGRL (right). Different from them, SPGRL maximizes the MI between the graph and feature embedding to explore the global structure information.

and feature embeddings. First, we construct a feature graph to provide a complementary view, which allows the feature information to propagate through the feature space, thus reinforcing the feature information and alleviating the uncertainty or error in the original graph.The original graph is often extracted from complex interaction systems that inevitably involve uncertain, redundant, wrong and missing connections [14]. Specifically, we use k-Nearest Neighbor (kNN) method to build the feature graph $\hat{\mathbf{A}}$, which could preserve high-order proximity. Then, the output embedding $\mathbf{Z^t}$ from A and $\mathbf{Z^f}$ from $\hat{\mathbf{A}}$ are obtained through GCN. They are refined by local node-level relation through contrastive loss. Finally, we maximize MI between embeddings and topology graph, which is theoretically equivalent to minimizing exchange reconstruction loss. Therefore, we reconstruct original graph with the embedding of feature graph and reconstruct feature graph with the embedding of original graph.

Our main contributions are summarized as follows:

- We propose to preserve the global structure information by maximizing the MI between topology graph and feature embeddings. Theoretical analysis shows that this can be achieved by exchange reconstruction.
- Our method explores the local node-level relation with the aid of feature graph. Feature view preserves high-order relations and helps eliminate the uncertainty or error in the original graph.
- Comprehensive experiments on benchmarks show the superior performance of our method compared to other state-of-the-art methods in semi-supervised node classification task. Our method also outperforms other mainstream methods even with very few labels and under noise perturbation.

II. RELATED WORKS

A. Graph Representation Learning

Due to the success of deep learning, graph neural network (GNN) approach has been developed. ChebNet [17] uses the Chebyshev polynomial approximation to optimize a general graph convolutional framework based on graph Laplacian. GCN [3] further simplifies the convolution operation using a localized first-order approximation. GAT [7] assigns different attention weights to different nodes in the neighborhood to better fuse node features. Demo-Net [18] builds a degree-specific GNN for the representation of nodes and graphs. MixHop [19] utilizes multiple powers of adjacency matrix to learn general mixing of neighborhood information. However, these methods only use a single topology graph for node aggregation. Some methods propose to solve this problem for better fusing node features by constructing feature graph. AMGCN [5] utilizes attention mechanism to merge embeddings extracted from topology graph and feature graph. However, these attention-based approaches are often computation expensive.

B. Self-supervised Learning

Recently, there have been several works focusing on self-supervised learning methods in the graph domain. M3S [20] utilizes a multi-stage, self-supervised learning approach to improve the generalization performance of GCN. GRACE [9] is a graph contrastive representation learning framework that seeks an optimal common representation. GCA [11] uses adaptive graph structure augmentation to construct a contrastive view and distinguishes the embeddings of the same node in two different views from the embedding of other nodes. SLAPS [10] solves the problem of underutilization of information in unsupervised learning by constructing a homogeneous node graph at graph level and contrasting it. DGI [12] first proposes the use of MI in the graph domain, which maximizes MI between hidden representation and a summary vector from a corrupted graph. But DGI's simple averaging readout function compromises global information. Unlike them, GMI [16] uses a discriminator to directly measure the MI between the input graph and output graph in terms of features and edges, not directly using local-global relationships. Due to these design flaws, they fail to take full advantage of the global graph information. Furthermore, most contrastive learning methods involve random destruction at nodes and edges. This could introduce noise to the original graph data and reduce the generalizability of the learned representations. Hence, there is much room to improve information utilization at the node level and graph level.

III. THE PROPOSED METHODOLOGY

The aim of our proposed method is to fully exploit potential correlations between graph structure and node attributes. In particular, not just capturing graph information from the original graph, we also exploit the feature view via feature graph. Ultimately we inherit rich representation information from feature graph view and topology graph view by maximizing global level MI.

A. Feature Extraction

We first outline the general setting of graph representation learning. A graph can be represented as $\mathbf{G} = \{\mathbf{A}, \mathbf{X}\}$, where $\mathbf{A} \in \mathbf{R^{N \times N}}$ is the adjacency matrix of $\mathbf{N}$ nodes

and $\mathbf{X} \in \mathbb{R}^{\mathbf{N}\times\mathbf{d}}$ is the node feature matrix, i.e., each node is described by a vector with $\mathbf{d}$ dimensions and belongs to one out of $\mathbf{M}$ classes. $\mathbf{A_{ij}} = 1$ represents that there is an edge between node $\mathbf{i}$ and $\mathbf{j}$, otherwise $\mathbf{A_{ij}} = 0$. In our study, we derive the feature graph $\hat{\mathbf{G}} = \left\{\hat{\mathbf{A}}, \mathbf{X}\right\}$, which shares the same $\mathbf{X}$ with $\mathbf{G}$, but has a different adjacency matrix. Therefore, topology graph and feature graph refer to $\mathbf{G}$ and $\hat{\mathbf{G}}$ respectively.

To represent the structure of nodes in the feature space, we build feature graph $\hat{\mathbf{G}}$ via kNN. First, a similarity matrix S is computed using the Cosine similarity, i.e., the similarity between node feature $\mathbf{x_i}$ and node feature $\mathbf{x_j}$ is $S_{ij} = \frac{\mathbf{x_i}\cdot\mathbf{x_j}}{\|\mathbf{x_i}\|\cdot\|\mathbf{x_j}\|}$. Then, for each node, we choose the top k nearest neighbors and establish edges. In this way, we construct the structure of the feature graph as $\hat{A}$.

To extract meaningful features from graph, we adopt GCN as our backbone. With the input graph $\mathbf{G}$, the $(l+1)$-th layer's output $H^{(l+1)}$ can be represented as: $H^{(l+1)} = ReLU(D^{-\frac{1}{2}}AD^{-\frac{1}{2}}H^{(l)}W^{(l)})$. where $ReLU$ is the Relu activation function ($ReLU(\cdot) = max(0, \cdot)$), $\mathbf{D}$ is the degree matrix of $\mathbf{A}$, $W^{(l)}$ is a layer-specific trainable weight matrix, $H^{(l)}$ is the output matrix in the l-th layer and $H^{(0)} = X$. In our study, we use two GCNs to exploit the information in topology and feature space. The output is denoted by $\mathbf{Z^t} = \left\{\mathbf{z_1^t}, \mathbf{z_2^t}, \cdots, \mathbf{z_N^t}\right\}$ and $\mathbf{Z^f} = \left\{\mathbf{z_1^f}, \mathbf{z_2^f}, \cdots, \mathbf{z_N^f}\right\}$, respectively.

B. Local Node-level Relation

Unlike previous graph contrastive learning models, SPGRL uses feature graph as a complementary view to capture local relation at the node-level. The feature graph characterizes high-order relations, thus the feature view encodes high-order structure information. Therefore, it provides complementary information to the original graph, which just describes the first-order relation and inevitably involves uncertainty or error. To learn a consistent representation, we uncover the local pairwise relations between nodes via a contrastive learning mechanism. Concretely, we treat $\mathbf{z_i^t}$ as a positive sample of $\mathbf{z_j^f}$ only when $\mathbf{i} = \mathbf{j}$ satisfies and $\mathbf{z_i^t}$ are negative samples of $\mathbf{z_j^f}$ for $\mathbf{i} \neq \mathbf{j}$, and vice versa. Then the loss can be formulated as:

$$L_{cr} = -\sum_{i=1}^{N} \log \frac{\exp(sim(\mathbf{z_i^t}, \mathbf{z_i^f}))}{\exp(sim(\mathbf{z_i^t}, \mathbf{z_i^f})) + \sum_{\mathbf{j=1,j\neq i}}^{N} \exp(sim(\mathbf{z_i^t}, \mathbf{z_j^f}))} - \sum_{i=1}^{N} \log \frac{\exp(sim(\mathbf{z_i^f}, \mathbf{z_i^t}))}{\exp(sim(\mathbf{z_i^f}, \mathbf{z_i^t})) + \sum_{\mathbf{j=1,j\neq i}}^{N} \exp(sim(\mathbf{z_i^f}, \mathbf{z_j^t}))}, \tag{1}$$

where $sim(\cdot, \cdot)$ is the Cosine function. Intuitively, the purpose of Eq.(1) is to make the representations of nodes within local neighborhood as close as possible and the representations of nodes from different groups as distinct as possible.

C. Global Graph-level Relation

Node contrastive method is not an effective way to attain global structural information in the topology graph. Existing approaches ignore the mutual corroboration effects of structures and attributes. The embedding of feature graph is expected to extract some relevant structure information from topology graph to improve the accuracy of downstream tasks. To this end, we propose to maximize the MI $I(\mathbf{Z^f}, \mathbf{A})$ between $\mathbf{Z^f}$ and whole topology graph $\mathbf{A}$ to preserve the structure information in topology graph. In addition, we also improve the embedding of topology graph $\mathbf{Z^t}$ by maximizing $I(\mathbf{Z^t}, \hat{\mathbf{A}})$ between $\mathbf{Z^t}$ and whole feature graph $\hat{\mathbf{A}}$.

Let's take $I(\mathbf{Z^f}, \mathbf{A})$ as an example to show the computation process. Mathematically, $I(\mathbf{Z^f}, \mathbf{A}) = \mathbb{E}_{p(\mathbf{Z^f},\mathbf{A})}\left[\log \frac{p(\mathbf{Z^f},\mathbf{A})}{p(\mathbf{Z^f})p(\mathbf{A})}\right]$. According to the relation between entropy and MI, we can decompose $I(\mathbf{Z^f}, \mathbf{A})$ as follows: $I(\mathbf{Z^f}, \mathbf{A}) = H(\mathbf{A}) - H(\mathbf{A}|\mathbf{Z^f})$, where $H(\mathbf{A}|\mathbf{Z^f}) = -\mathbb{E}_{p(\mathbf{Z^f},\mathbf{A})}\left[\log p(\mathbf{A}|\mathbf{Z^f})\right]$ is the conditional entropy, and $H(\mathbf{A})$, the entropy of $\mathbf{A}$, is irrelevant to $\mathbf{Z^f}$. Hence, maximizing $I(\mathbf{Z^f}, \mathbf{A})$ is equivalent to maximizing $-H(\mathbf{A}|\mathbf{Z^f})$. However, the computation of $H(\mathbf{A}|\mathbf{Z^f})$ is intractable due to unknown of the condition distribution $p(\mathbf{A}|\mathbf{Z^f})$.

We assume $q_\phi(\mathbf{A}|\mathbf{Z^f})$ is a variational approximation to $p(\mathbf{A}|\mathbf{Z^f})$. Since $KL(p(\mathbf{A}|\mathbf{Z^f})||q_\phi(\mathbf{A}|\mathbf{Z^f})) \geq 0$, we can derive that: $\mathbb{E}_{p(\mathbf{Z^f},\mathbf{A})}\left[\log p(\mathbf{A}|\mathbf{Z^f})\right] \geq \mathbb{E}_{p(\mathbf{Z^f},\mathbf{A})}\left[\log q_\phi(\mathbf{A}|\mathbf{Z^f})\right]$. Hence, $\mathbb{E}_{p(\mathbf{Z^f},\mathbf{A})}\left[\log q_\phi(\mathbf{A}|\mathbf{Z^f})\right]$ is the lower bound of $\mathbb{E}_{p(\mathbf{Z^f},\mathbf{A})}\left[\log p(\mathbf{A}|\mathbf{Z^f})\right]$. Specifically, $q_\phi(\mathbf{A}|\mathbf{Z^f})$ can be regarded as the decoder function whose equation is as follows: $q_\phi(\mathbf{A}|\mathbf{Z^f}) = \prod_{i=1}^{N}\prod_{j=1}^{N} q_\phi\left(\mathbf{A_{ij}} \mid \mathbf{z_i^f}, \mathbf{z_j^f}\right)$, where the probability of an edge existing between two nodes is: $q_\phi\left(\mathbf{A_{ij}} = 1 \mid \mathbf{z_i^f}, \mathbf{z_j^f}\right) = \text{sigmoid}\left(\mathbf{z_i^{f^T}}\mathbf{z_j^f}\right)$.

Above optimization objective of maximizing $I(\mathbf{Z^f}, \mathbf{A})$ is equivalent to:

$$L_{re}^{\mathbf{A}} = E_{p(\mathbf{Z^f},\mathbf{A})}[\log q_\phi(\mathbf{A} \mid \mathbf{Z^f})]. \tag{2}$$

Likewise, we can obtain a similar objective of maximizing $I(\mathbf{Z^t}, \hat{\mathbf{A}})$ as follows:

$$L_{re}^{\hat{\mathbf{A}}} = E_{p(\mathbf{Z^t},\hat{\mathbf{A}})}[\log q_\phi(\hat{\mathbf{A}} \mid \mathbf{Z^t})]. \tag{3}$$

To summarize, we propose the exchange-reconstruction mechanism to maximize $I(\mathbf{Z^f}, \mathbf{A})$ and $I(\mathbf{Z^t}, \hat{\mathbf{A}})$ between the embeddings and graph structures. Then the global MI loss can be formulated as:

$$L_{re} = L_{re}^{\mathbf{A}} + L_{re}^{\hat{\mathbf{A}}}. \tag{4}$$

D. Node Classification

Ideally, $\mathbf{Z^t}$ and $\mathbf{Z^f}$ should be close to each other. To preserve the information from feature graph and topology graph, $\mathbf{Z^t}$ and $\mathbf{Z^f}$ are concatenated as the consensus representation R [13]. Then we use $\mathbf{R}$ for semi-supervised classification, which is realized through a linear transformation and a softmax function. $\mathbf{B}$ and $\mathbf{a}$ are weights and bias of the linear layer, respectively. $\mathbf{Y'}$ is the prediction result and $\mathbf{Y'_{ij}}$ is the probability of node $\mathbf{i}$ belonging to class $\mathbf{j}$, $\mathbf{Y'} = \text{softmax}(\mathbf{B}\cdot\mathbf{R}+\mathbf{a})$. Suppose there are $\mathcal{T}$ nodes with labels in the training set. We adopt cross-

entropy to measure the difference between prediction label $\mathbf{Y'_{ij}}$ and ground truth label $\mathbf{Y_{ij}}$, i.e.,

$$L_{cl} = -\sum_{\mathbf{i=1}}^{\mathcal{T}} \sum_{\mathbf{j=1}}^{\mathbf{M}} \mathbf{Y_{ij}} \ln \mathbf{Y'_{ij}}. \tag{5}$$

Finally, by combining L_{cl}, L_{re} and L_{cr}, the overall loss function of our SPGRL model can be represented as:

$$L = L_{cl} + \alpha L_{re} + \beta L_{cr}, \tag{6}$$

where α and β are trade-off hyper-parameters. The parameters of the whole framework are updated via backpropagation. The detailed description of our algorithm is provided in Algorithm 1.

Algorithm 1: The proposed algorithm SPGRL

Input: Node feature matrix $\mathbf{X}$; original graph adjacency matrix $\mathbf{A}$; node label matrix $\mathbf{Y}$; maximum number of iterations η

Compute the feature graph topological structure $\hat{\mathbf{A}}$ according to $\mathbf{X}$ by running kNN algorithm.

for $it = 1$ **to** η **do**

  $\mathbf{Z^t} = GCN(\mathbf{A}, \mathbf{X})$

  $\mathbf{Z^f} = GCN'(\hat{\mathbf{A}}, \mathbf{X})$ `// embeddings of two graphs`

  $\mathbf{Z^t}$ and $\mathbf{Z^f}$ interact with local node-level information.

  $q_\phi(\hat{\mathbf{A}}|\mathbf{Z^t}) = Decoder(\mathbf{Z^t})$

  $q_\phi(\mathbf{A}|\mathbf{Z^f}) = Decoder'(\mathbf{Z^f})$ `// reconstructing two graphs`

  $q_\phi(\hat{\mathbf{A}}|\mathbf{Z^t})$ constrained by $\hat{\mathbf{A}}$, $q_\phi(\mathbf{A}|\mathbf{Z^f})$ constrained by $\mathbf{A}$

  Calculate the overall loss with Eq.(6)

  Update all parameters of framework according to the overall loss

end

Predict the labels of unlabeled nodes based on the trained framework.

Output: Classification result $\mathbf{Y'}$

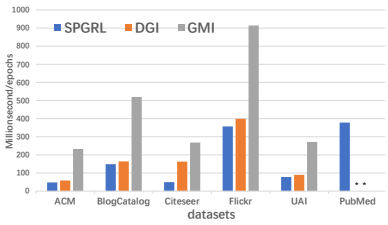

Fig. 2. Averaged time cost per epoch of SPGRL, DGI and GMI for six datasets. (*) indicates out-of-memory error and vertical axis is in log-scale.

IV. EXPERIMENT

A. Setup

We use six commonly used datasets to evaluate the effectiveness of our method [13]. The experiments are implemented in the PyTorch platform using an Intel(R) Xeon(R) Gold 5218 CPU, and GeForce RTX 3090 24G GPU. Technically, two layers GCN is built and we train our model by utilizing the Adam optimizer with learning rate ranging from 0.0001 to 0.0005. In order to prevent over-fitting, we set the dropout rate to 0.5. In addition, we set weight decay $\in \{1e-4, \cdots, 5e-3\}$ and $k \in \{2, \cdots, 20\}$ for kNN graph. For fairness, we follow Wang *et al.* [5] and select 20, 40, 60 nodes per class for training and 1000 nodes for testing. For example, there are 6 types of nodes in Citeseer, therefore we train our model on training set with 120/240/360 nodes, corresponding to label rate of 3.61%, 7.21%, 10.82%, respectively. Two popular metrics are applied to quantitatively evaluate the semi-supervised node classification: Accuracy (ACC) and F1-Score (F1). We repeatedly train and test our model for five times with the same partition of dataset and then report the average of ACC and F1.

We choose some representative methods to compare, including DeepWalk [21], LINE [22], ChebNet [17], GCN [3], kNN-GCN [5], GAT [7], Demo-Net [18], MixHop [19] and AMGCN [5], DGI [12], GRACE [9], GMI [16], SCRL [13], SLAPS [10] and GCA [11].

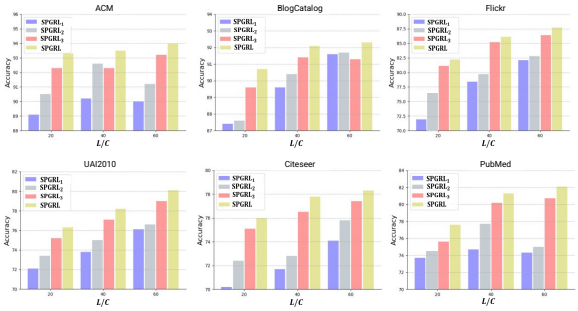

Fig. 3. The classification accuracy (%) of SPGRL and its variants on six datasets.

B. Node Classification Results

The results of experiments are summarized in Table I, where the best performance is highlighted in boldface. Some results are directly taken from [5], [13]. We have the following findings:

(1) It can be seen that our proposed method boosts the performance of STOA methods across most evaluation metrics on six datasets, which proves its effectiveness. Particularly, compared with other optimal performance, SPGRL achieves a maximum improvement of 4.90% for ACC and 3.84% for F1 on UAI2010. This illustrates that our proposed model can effectively fuse topological structure and feature.

(2) Our SPGRL achieves much better performances than DGI and GMI on all of the metrics. This can be explained by the fact that our method fully exploits the global structure via the MI maximization between graph structure and embedding.

(3) In most cases, SPGRL produces better performance than SCRL [13], SLAPS [10], and GCA [11], which were published in 2021. This verifies the advantage of our approach.

(4) On some occasions, feature graph produces better result than original graph. For example, on BlogCatalog, Flickr, and UAI2010, kNN-GCN beats GCN. This confirms that incorporating feature graph into our framework can avoid uncertainty or error information in the original graph in many cases.

To verify the efficiency of SPGRL, we report the averaged training time per epoch when training SPGRL, DGI and GMI in Fig.2. It can be seen that SPGRL always costs much less time than others.

TABLE I
NODE CLASSIFICATION RESULTS(%). L/C REFERS TO THE NUMBER OF LABELED NODES PER CLASS.

Dataset	**ACM**						**BlogCatalog**					
L/C	20		40		60		20		40		60	
Label Rate	1.98%		3.97%		5.95%		2.31%		4.62%		6.93%	
Metrics	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1
DeepWalk [21]	62.69	62.11	63.00	61.88	67.03	66.99	38.67	34.96	50.80	48.61	55.02	53.36
LINE [22]	41.28	40.12	45.83	45.79	50.41	49.92	58.75	57.75	61.12	60.72	64.53	63.81
ChebNet [17]	75.24	74.86	81.64	81.26	85.43	85.26	38.08	33.39	56.28	53.86	70.06	68.37
GCN [3]	87.80	87.82	89.06	89.00	90.54	90.49	69.84	68.73	71.28	70.71	72.66	71.80
kNN-GCN [5]	78.52	78.14	81.66	81.53	82.00	81.95	75.49	72.53	80.84	80.16	82.46	81.90
GAT [7]	87.36	87.44	88.60	88.55	90.40	90.39	64.08	63.38	67.40	66.39	69.95	69.08
Demo-Net [18]	84.48	84.16	85.70	84.83	86.55	84.05	54.19	52.79	63.47	63.09	76.81	76.73
MixHop [19]	81.08	81.40	82.34	81.13	83.09	82.24	65.46	64.89	71.66	70.84	77.44	76.38
DGI [12]	90.48	90.40	90.97	90.88	90.94	90.79	64.59	63.58	65.09	64.15	65.90	65.00
GRACE [9]	89.04	89.00	89.46	89.36	91.08	91.03	76.56	75.56	76.66	75.88	77.66	77.08
AMGCN [5]	90.40	90.43	90.76	90.66	91.42	91.36	81.89	81.36	84.94	84.32	87.30	86.94
GMI [16]	90.22	90.00	90.68	90.64	91.48	91.45	66.46	39.2	68.01	40.42	72.59	43.24
SCRL [13]	91.82	91.79	92.06	92.04	92.82	92.80	90.22	89.89	90.26	89.90	91.58	90.76
SLAPS [10]	65.32	60.00	55.46	47.73	60.13	52.56	87.80	87.34	88.50	87.57	89.50	89.22
GCA [11]	88.39	8879	91.95	90.99	91.75	90.79	80.51	81.28	84.89	84.04	86.34	86.19
SPGRL	**93.30**	**93.27**	**93.50**	**93.48**	**94.00**	**93.98**	**90.70**	**90.12**	**92.10**	**91.34**	**92.30**	**92.13**
Dataset	**Flickr**						**UAI2010**					
L/C	20		40		60		20		40		60	
Label Rate	2.38%		4.75%		7.13%		12.39%		24.78%		37.17%	
Metrics	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1
DeepWalk [21]	24.33	21.33	28.79	26.90	30.10	27.28	42.02	32.92	51.26	46.01	54.37	44.43
LINE [22]	33.25	31.19	37.67	37.12	38.54	37.77	43.47	37.01	45.37	39.62	51.05	43.76
ChebNet [17]	23.26	21.27	35.10	33.53	41.70	40.17	50.02	33.65	58.18	38.80	59.82	40.60
GCN [3]	41.42	39.95	45.48	43.27	47.96	46.58	49.88	32.86	51.80	33.80	54.40	32.14
kNN-GCN [5]	69.28	70.33	75.08	75.40	77.94	77.97	66.06	52.43	68.74	54.45	71.64	54.78
GAT [7]	38.52	37.00	38.44	36.94	38.96	37.35	56.92	39.61	63.74	45.08	68.44	48.97
Demo-Net [18]	34.89	33.53	46.57	45.23	57.30	56.49	23.45	16.82	30.29	26.36	34.11	29.03
MixHop [19]	39.56	40.13	55.19	56.25	64.96	65.73	61.56	49.19	65.05	53.86	67.66	56.31
DGI [12]	34.95	33.1	34.98	33.07	35.51	34.37	33.26	11.86	32.55	9.29	32.44	9.37
GRACE [9]	49.42	48.18	53.64	52.61	55.67	54.61	65.54	48.38	66.67	49.50	68.68	51.51
AMGCN [5]	75.26	74.63	80.06	79.36	82.10	81.81	70.10	55.61	73.14	64.88	74.40	65.99
GMI [16]	49.17	28.43	52.74	30.94	53.78	31.50	60.69	46.75	63.14	49.10	64.73	44.36
SCRL [13]	79.52	78.89	84.23	84.03	84.54	84.51	72.90	57.80	74.58	67.40	74.90	67.54
SLAPS [10]	72.20	72.48	79.00	78.90	76.20	76.50	46.82	41.60	34.62	25.28	62.51	51.81
GCA [11]	63.44	63.26	63.90	64.60	64.43	64.64	72.55	56.97	73.27	54.55	73.60	56.00
SPGRL	**82.20**	**81.24**	**86.20**	**85.93**	**87.10**	**85.97**	**76.30**	**61.49**	**78.20**	**68.73**	**79.80**	**71.38**
Dataset	**Citeseer**						**PubMed**					
L/C	20		40		60		20		40		60	
Label Rate	3.61%		7.21%		10.82%		0.30%		0.61%		0.91%	
Metrics	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1
DeepWalk [21]	43.47	38.09	45.15	43.18	48.86	48.01	-	-	-	-	-	-
LINE [22]	32.71	31.75	33.32	32.42	35.39	34.37	-	-	-	-	-	-
ChebNet [17]	69.80	65.92	71.64	68.31	73.26	70.31	74.20	73.51	76.00	74.92	76.51	75.83
GCN [3]	70.30	67.50	73.10	69.70	74.48	71.24	79.00	78.45	79.98	79.17	80.06	79.65
kNN-GCN [5]	61.35	58.86	61.54	59.33	62.38	60.07	71.62	71.92	74.02	74.09	74.66	75.18
GAT [7]	72.50	68.14	73.04	69.58	74.76	71.60	-	-	-	-	-	-
Demo-Net [18]	69.50	67.84	70.44	66.97	71.86	68.22	-	-	-	-	-	-
MixHop [19]	71.40	66.96	71.48	67.40	72.16	69.31	-	-	-	-	-	-
DGI [12]	71.24	67.05	71.26	67.75	73.92	70.26	-	-	-	-	-	-
GRACE [9]	71.70	68.14	72.38	68.74	74.20	70.73	79.50	79.33	80.32	79.64	80.24	80.33
AMGCN [5]	73.10	68.42	74.70	69.81	75.56	70.92	76.18	76.86	77.14	77.04	77.74	77.09
GMI [16]	71.24	67.1	73.1	68.57	73.96	70.25	-	-	-	-	-	-
SCRL [13]	73.62	69.78	75.08	70.68	75.96	72.84	79.62	78.88	80.74	80.24	81.03	80.55
SLAPS [10]	70.50	67.23	72.10	69.15	73.00	69.80	71.70	72.29	71.60	71.56	70.60	71.16
GCA [11]	71.39	68.46	72.96	68.02	73.92	69.10	**82.00**	**81.50**	**82.59**	**82.43**	82.03	81.75
SPGRL	**75.90**	**70.98**	**77.40**	**73.75**	**78.30**	**73.98**	77.60	76.98	81.20	81.01	**82.10**	**81.94**

TABLE II
CLASSIFICATION ACCURACY WITH LOW LABEL RATES.

Datasets	Citeseer				PubMed		
L/C	3	6	12	18	2	3	7
Label Rate	0.5%	1%	2%	3%	0.03%	0.05%	0.10%
ChebNet [17]	19.7	59.3	62.1	66.8	55.9	62.5	69.5
GCN [3]	33.4	46.5	62.6	66.9	61.8	68.8	71.9
GAT [7]	45.7	64.7	69.0	69.3	65.7	69.9	72.4
DGI [12]	60.7	66.9	68.1	69.8	60.2	68.4	70.7
M3S [20]	56.1	62.1	66.4	70.3	59.2	64.4	70.5
GRACE [9]	55.4	59.3	63.4	67.8	64.4	67.5	72.3
AMGCN [5]	60.2	65.7	68.5	70.2	60.5	62.4	70.8
SCRL [13]	62.4	67.3	69.8	73.3	67.9	71.9	73.4
GCA [11]	62.6	63.4	62.7	60.8	70.1	73.2	75.8
SPGRL	**64.3**	**68.4**	**71.7**	**74.7**	**70.2**	**73.4**	**76.7**

C. Ablation Study

To validate the effectiveness of different components in our model, we compare SPGRL with its three variants on all datasets.

- **SPGRL**$_1$: SPGRL without L_{cr} and L_{re} to show the impact of local and global structure.
- **SPGRL**$_2$: SPGRL without L_{re} to show the effect of global structure preserving.
- **SPGRL**$_3$: SPGRL with traditional reconstruction, i.e., $q_\phi(\mathbf{A}|\mathbf{Z^t})$ and $q_\phi(\hat{\mathbf{A}}|\mathbf{Z^f})$, to demonstrate the benefit of exchange-reconstruction.

According to Fig.3, we can draw the following conclusions: (1) The results of SPGRL are consistently better than all variants, indicating the rationality of our model. (2) Both local and global structure information are crucial to representation learning. (3) Exchange reconstruction is beneficial by removing some redundant information.

D. Few Labeled Classification

To further investigate the capability of SPGRL in dealing with scarce supervision data, we conduct experiments when the number of labeled examples is extremely small. Taking Citeseer and PubMed for example, we select a small set of labeled examples for model training [23]. Specifically, for Citeseer, we select 3, 6, 12, 18 nodes per class, corresponding to label rates: 0.5%, 1%, 2%, and 3%; for PubMed, we select 2, 3, 7 nodes per class, corresponding to three label rates: 0.03%, 0.05% and 0.10%. To make a fair comparison, we report mean classification accuracy of 10 runs.

From Table II, we can observe that SPGRL outperforms all STOA approaches. For example, SPGRL improves AMGCN,

SCRL, GCA by 5.87%, 1.91%, and 4.40% on average. Particularly, the accuracy of GCN, ChebNet, and GAT decline severely when the label rate is very low, especially on 0.5% Citeseer, due to insufficient propagation of label information. By contrast, self-supervised/contrastive approaches are obviously much better because they additionally exploit supervisory signals. Though GCA outperforms SPGRL in most cases of Pubmed dataset in Table I, its performance is worse than our method at low label rate. Thus, fully exploring structure information could alleviate the reliance of label to some extent.

E. Experiments with Noise Perturbation

Many recent studies have found that GCN is vulnerable to noise perturbation on node features or graph structure.Hence, it is necessary to evaluate the robustness of our method. We perturb node features by injecting independent Gaussian noise. Consequently, our built feature graph is also corrupted. Note that it is computationally expensive to perturb structure and it behaves similarly to feature perturbation to some extent [24]. Therefore, there is no need to corrupt original graph structure $\mathbf{A}$ in our setting. Specifically, we add Gaussian noise to input features: $\mathbf{X} \leftarrow \mathbf{X} + \mathcal{N}(0, \sigma^2)$, where σ is the variance of Gaussian noise. We compare to a few closely relevant methods, including GFNN [25], which employs low-pass filtering to remove noise.

Table III shows results with $\sigma = 1$ on ACM dataset. We also test with $\sigma \in \{0.01, 0.02, ..., 2.0\}$ in Fig.4. The results show that SPGRL still performs the best in most scenarios. Its robustness could be explained by the fact that we extract more relevant information from the original graph by maximizing the MI between it and the embeddings, which alleviates the negative influence of noise perturbation.

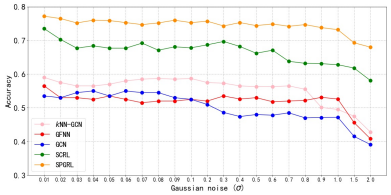

Fig. 4. Accuracy of SPGRL under different σ on ACM dataset (L/C=20).

TABLE III
NODE CLASSIFICATION WITH GAUSSIAN NOISE PERTURBATION ($\sigma = 1.0$).

Dataset	Metrics	L/C	SPGRL	SCRL [13]	GFNN [25]	GCN [3]	*k*NN-GCN [5]
ACM	ACC	20	**73.2**	62.8	52.6	47.2	49.5
		40	**78.1**	75.2	50.1	52.1	57.4
		60	**86.6**	80.6	56.4	57.2	58.8
BlogCatalog	ACC	20	**80.3**	75.0	62.4	55.1	56.1
		40	**86.2**	77.9	47.3	57.7	63.2
		60	**89.3**	78.3	53.4	57.1	61.4
UAI2010	ACC	20	**72.8**	69.4	32.8	49.9	52.0
		40	**73.3**	73.3	32.2	53.0	55.0
		60	76.8	**76.9**	29.5	57.5	58.9
Flickr	ACC	20	**65.3**	54.2	21.3	27.5	35.5
		40	**65.6**	61.9	20.3	31.3	29.4
		60	**74.1**	73.5	24.3	34.4	32.3
Citeseer	ACC	20	45.3	**51.3**	36.1	34.4	36.6
		40	**59.8**	59.7	40.9	43.4	41.6
		60	**66.2**	63.2	46.0	45.7	47.8

V. CONCLUSION

In this paper, we propose a framework to preserve the local-global structure information during graph embedding. This is mainly realized by maximizing MI between topological structure and feature representation, which is further converted to exchange reconstruction according to our theoretical derivation. Comprehensive experiments verify the effectiveness, efficiency, and robustness of our approach in different scenarios.

VI. ACKNOWLEDGEMENTS

This work was supported by the Natural Science Foundation of China under Grant 62276053.

REFERENCES

[1] Z. Lin, Z. Kang, L. Zhang, and L. Tian, "Multi-view attributed graph clustering," *IEEE Transactions on Knowledge and Data Engineering*, 2021.
[2] L. Liu, Z. Kang, J. Ruan, and X. He, "Multilayer graph contrastive clustering network," *Information Sciences*, 2022.
[3] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017.
[4] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger, "Simplifying graph convolutional networks," in *ICML*, 2019.
[5] X. Wang, M. Zhu, D. Bo, P. Cui, C. Shi, and J. Pei, "Am-gcn: Adaptive multi-channel graph convolutional networks," in *ACM SIGKDD*, 2020.
[6] Z. Ma, Z. Kang, G. Luo, L. Tian, and W. Chen, "Towards clustering-friendly representations: Subspace clustering via graph filtering," in *ACM MM*, 2020.
[7] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph Attention Networks," *ICLR*, 2018.
[8] E. Pan and Z. Kang, "Multi-view contrastive graph clustering," *NeurIPS*, 2021.
[9] Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, and L. Wang, "Deep Graph Contrastive Representation Learning," in *ICML Workshop on Graph Representation Learning and Beyond*, 2020.
[10] B. Fatemi, L. El Asri, and S. M. Kazemi, "Slaps: Self-supervision improves structure learning for graph neural networks," *NeurIPS*, 2021.
[11] Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, and L. Wang, "Graph contrastive learning with adaptive augmentation," in *WWW*, 2021.
[12] P. Veličković, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, "Deep graph infomax," in *ICLR*, 2019.
[13] C. Liu, L. Wen, Z. Kang, G. Luo, and L. Tian, "Self-supervised consensus representation learning for attributed graph," in *ACM MM*, 2021.
[14] R. Wang, S. Mou, X. Wang, W. Xiao, Q. Ju, C. Shi, and X. Xie, "Graph structure estimation neural networks," in *WWW*, 2021.
[15] D. Luo, W. Cheng, W. Yu, B. Zong, J. Ni, H. Chen, and X. Zhang, "Learning to drop: Robust graph neural network via topological denoising," in *WSDM*, 2021.
[16] Z. Peng, W. Huang, M. Luo, Q. Zheng, Y. Rong, T. Xu, and J. Huang, "Graph representation learning via graphical mutual information maximization," in *WWW*, 2020.
[17] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *NeurIPS*, 2016.
[18] J. Wu, J. He, and J. Xu, "Net: Degree-specific graph neural networks for node and graph classification," in *ACM SIGKDD*, 2019.
[19] S. Abu-El-Haija, B. Perozzi, A. Kapoor, N. Alipourfard, K. Lerman, H. Harutyunyan, G. Ver Steeg, and A. Galstyan, "Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing," in *ICML*, 2019.
[20] K. Sun, Z. Lin, and Z. Zhu, "Multi-stage self-supervised learning for graph convolutional networks on graphs with few labeled nodes," in *AAAI*, 2020.
[21] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *ACM SIGKDD*, 2014.
[22] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *WWW*, 2015.
[23] Q. Li, Z. Han, and X.-M. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *AAAI*, 2018.
[24] W. Jin, Y. Ma, X. Liu, X. Tang, S. Wang, and J. Tang, "Graph structure learning for robust graph neural networks," in *ACM SIGKDD*, 2020.
[25] H. NT and T. Maehara, "Revisiting graph neural networks: All we have is low-pass filters," 2019.