GRLC: Graph Representation Learning With Constraints

Liang Peng, Yujie Mo, Jie Xu, Jialie Shen, *Senior Member, IEEE*, Xiaoshuang Shi, Xiaoxiao Li, *Member, IEEE*, Heng Tao Shen, *Fellow, IEEE*, and Xiaofeng Zhu, *Senior Member, IEEE*

***Abstract*— Contrastive learning has been successfully applied in unsupervised representation learning. However, the generalization ability of representation learning is limited by the fact that the loss of downstream tasks (e.g., classification) is rarely taken into account while designing contrastive methods. In this article, we propose a new contrastive-based unsupervised graph representation learning (UGRL) framework by 1) maximizing the mutual information (MI) between the semantic information and the structural information of the data and 2) designing three constraints to simultaneously consider the downstream tasks and the representation learning. As a result, our proposed method outputs robust low-dimensional representations. Experimental results on 11 public datasets demonstrate that our proposed method is superior over recent state-of-the-art methods in terms of different downstream tasks. Our code is available at *https://github.com/LarryUESTC/GRLC*.**

***Index Terms*— Data mining, graph neural networks, graph representation learning, machine learning.**

I. Introduction

UNSUPERVISED graph representation learning (UGRL) has received considerable attention in real applications, such as communication networks and citation networks, due to discovering discriminative representations without the help of label information [1], [2], [3], [4], [5]. To do this, UGRL via contrastive learning (C-UGRL) is designed to either minimize uni-modal losses or maximize the mutual information (MI) between the input information and its related information for representation learning. In real applications, uni-modal losses (e.g., the mean squared error and the cross-entropy loss) are inefficient for learning discriminative representations as they focus on every detail of the reconstruction process [6]. As their alternative, C-UGRL is popularly designed to efficiently achieve the MI maximization [i.e., $I(\mathbf{a}; \mathbf{b})$] between the representations/embeddings of $\mathbf{a}$ (i.e., input information) and $\mathbf{b}$ (i.e., related information of $\mathbf{a}$), for generating discriminative embeddings [7], [8]. The key components of the C-UGRL method include the encoder (e.g., graph convolutional network (GCN) [9]), the related information, and the MI maximization.

The related information is the key element for the MI maximization to generate discriminative embeddings [10] and is often generated by pretext tasks such as the modality different from the input, parts of inputs, and transformation of the inputs [11]. Early C-UGRL methods prefer to use "part of inputs" as the related information. For example, Velickovic et al. [12] regarded the local regions of the input (i.e., local information) as the related information because the local information and the global information (i.e., graph summary) can form cross-scale MI. Zhu et al. [13] and Tian et al. [14] regarded the transformation of the inputs as the related information. Recently, Hassani and Khasahmadi [15] considered multi-view information as the related information. Obviously, the definitions of the related information in the previous methods are becoming more complex, aiming at empirically boosting the effectiveness of the C-UGRL. As a result, they make the C-UGRL inflexible in terms of theoretical analysis and practical application.

Directly computing the MI is arduous in deep learning models and negative embeddings [10], [12], [16]. Specifically, C-UGRL needs to ensure positive pair (i.e., $\mathbf{z}$ and $\mathbf{z}^+$) close and negative pair (i.e., $\mathbf{z}$ and $\mathbf{z}^-$) far away in the embedding space, where $\mathbf{z}$, $\mathbf{z}^+$, and $\mathbf{z}^-$, respectively, stand for the anchor embedding, the positive embedding, and the negative embedding. Previous C-UGRL methods (e.g., [10], [14], [17], [18]) heavily relied on extensively empirical experiments to improve the effectiveness of the C-UGRL, but they paid little attention to theoretical connections between contrastive learning and downstream tasks while designing contrastive methods.

In this article, we propose a new C-UGRL method, referred as **G**raph **R**epresentation **L**earning with **C**onstraints (***GRLC***), shown in Fig. 1, to address the above limitations. First, we design a new C-UGRL architecture to maximize the MI

Manuscript received 17 March 2022; revised 25 November 2022; accepted 12 December 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFA1004100, in part by the National Natural Science Foundation of China under Grant 61876046, in part by the Medico-Engineering Cooperation Funds from University of Electronic Science and Technology of China under Grant ZYGX2022YGRH009 and Grant ZYGX2022YGRH014, and in part by the Guangxi "Bagui" Teams for Innovation and Research, China. *(Liang Peng and Yujie Mo contributed equally to this work.) (Corresponding author: Xiaofeng Zhu.)*

Liang Peng, Yujie Mo, Jie Xu, Xiaoshuang Shi, and Heng Tao Shen are with the Center for Future Media and School of Computer Science and Technology, University of Electronic Science and Technology of China, Chengdu 611731, China.

Jialie Shen is with the Department of Computer Science, City, University of London, EC1V 0HB London, U.K.

Xiaoxiao Li is with the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC V6T 1Z4, Canada.

Xiaofeng Zhu is with the School of Computer Science and Technology, University of Electronic Science and Technology of China, Chengdu 611731, China, and also with the Guangxi Academy of Sciences, Nanning 530007, China (e-mail: seanzhuxf@gmail.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.2022.3230979.

Digital Object Identifier 10.1109/TNNLS.2022.3230979

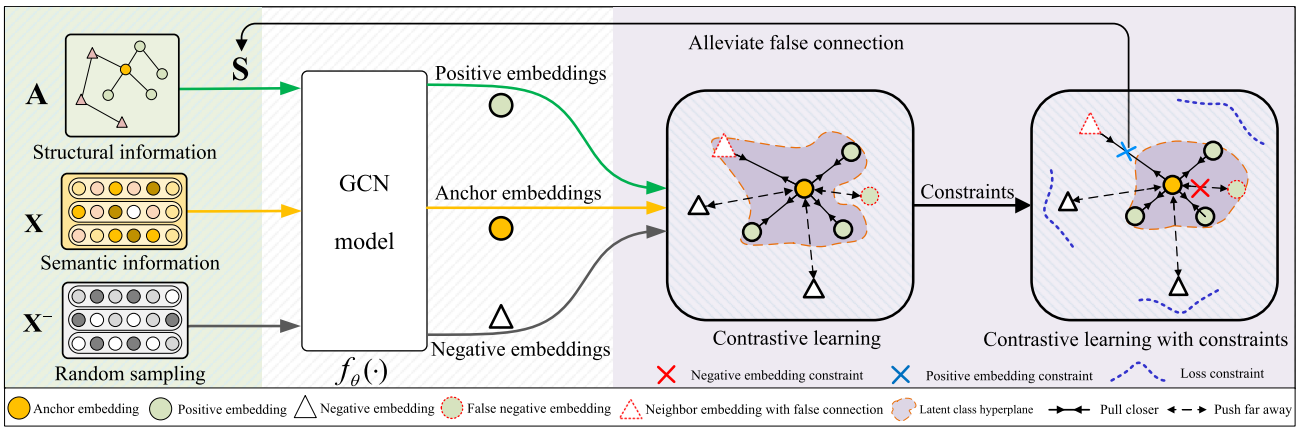

Fig. 1. Flowchart of the proposed **GRLC** method. Given the original feature **X** (i.e., the semantic information) and the graph structure **A** (i.e., the structural information), **GRLC** is first proposed to maximize the MI between the semantic information and the structural information, and then **GRLC** designs a *negative embedding constraint* to dynamically avoid class collisions and a *positive embedding constraint* to dynamically update the structural information for mitigating the impact of false connections. Finally, **GRLC** investigates a *loss constraint* to keep the distance between the anchor embedding and the positive embedding finite.

between the semantic information and the structural information of the graph-structure data. The motivation is that both the semantic information and the structural information are critical and essential information for learning the graph-structure data, and have been shown to result in discriminative embeddings in the GCN model [9]. Second, we theoretically investigate three constraints to reduce the gap between the unsupervised representation learning and the downstream task, considering the fact that the final goal of representation learning is to serve downstream tasks. Specifically, the *negative embedding constraint* adaptively adjusts the weights of negative embeddings, aiming at avoiding that anchor embeddings and negative embeddings belong to the same latent class, i.e., false negative embeddings. The *positive embedding constraint* alternatively updates structural information, aiming at avoiding that positive embeddings aggregate information from neighbors which have different latent classes from anchor embedding. Furthermore, we transfer the MI maximization to a triplet loss plus a *loss constraint*, aiming at guaranteeing positive embeddings close to their corresponding anchor embeddings as well as avoiding the increase of the intraclass deviation induced by the large representation norm. Compared to previous works, the main contributions of the proposed idea are listed as follows.

1) We design a novel and effective graph representation learning method (**GRLC**), which maximizes the MI between the semantic information and the structural information of the graph data.
2) **GRLC** contains three constraints to tackle the realistic issue of gaps between contrastive graph representation learning and downstream tasks.
3) We conduct theoretical discussion to support our motivation and justify in theory the property of **GRLC** which is the low gap between contrastive graph representation learning and downstream tasks.

II. Related Works

A. Self-Supervised Learning

Self-supervised learning has recently emerged in the domain of deep learning as a strong unsupervised representation learning approach [11], [19]. Unlike previous methods of traditional unsupervised learning [2], [20], [21], self-supervised learning takes advantage of maximizing the MI on multi-view data to extract the embeddings of data [22]. On this basic, previous works have shown that self-supervised learning has powerful embedding ability on many kinds of tasks with unlabeled data, such as image classification [23], graph classification [24], [25], and disease diagnosis [19], [26]. According to the definition of multi-view data and the algorithm of maximizing the mutual information, existing self-supervised learning methods can be broadly classified into three groups, i.e., contrastive learning, reconstruction learning, and similarity-based self-supervised learning [11], [12], [19].

B. Contrastive Learning

As one effective self-supervised learning teleology, contrastive learning has recently attracted attention for its success in unsupervised representation learning in the field of computer vision (CV) [10], [27], [28] and natural language processing (NLP) [29], [30]. In particular, it learns new representations for downstream tasks by utilizing human-defined positive and negative samples as self-supervised information. For example, SimCLR [10] and MoCo [28] significantly improve the quality of the representation by designing appropriate data augmentation techniques. SimCSE [16] takes an input sentence and predicts itself in a contrastive objective, with only standard dropout used as noise, and performs on par with previous supervised counterparts. BERT [29] shows huge improvement on the NLP downstream tasks by pretraining. Moreover, a number of recent works have attempted to theoretically explain the success of contrastive learning. Saunshi et al. [31] presented a theoretical guarantee for contrastive learning that establishes the connection between contrastive learning and downstream tasks. Wang and Isola [32] provided a theoretical analysis on alignment and uniformity of representations. Tsai et al. [33] provided theoretical understanding based on causality. Some recent works [11], [34] took a deeper look at what types of pretext are more beneficial for downstream tasks at the theoretical level. But little work in the

field of graph representation learning has used these theories to design better methods for contrastive learning.

C. Contrastive Unsupervised Graph Representation Learning

Powerful contrastive learning has recently been developed in graph representation learning field. Existing C-UGRL methods can be broadly classified into two groups based on the pretext task i.e., contrasting with the cross-scale and contrasting with the same-scale.

The cross-scale contrasting methods enhance the similarity between the local representation and the different scale representation (i.e., global representation or cluster representation). For example, deep graph infomax (DGI) [12] employs the local–global contrastive learning to contrast the local node representations and the global graph representations. InfoGraph [35] extends the node-level DGI [12] to learn graph-level representations. Graph InfoClust (GIC) [36] maximizes the agreement between the local node representations and their corresponding cluster centroids. Multi-view representation learning on graphs (MVGRL) [15] proposes a contrastive multi-view representation learning method by contrasting local and global embeddings from two views.

The same-scale contrasting methods enhance the similarity between representations within the same scale. These methods generally rely on different data augmentation, i.e., different types of transformations to the graph structure or the node feature. For example, deep graph contrastive representation learning (GRACE) [13] generates two views by masking edges and node features, and then pulls the representations of the same nodes in two views close. GraphCL [18] uses the edge perturbation to randomly add or subtract edges to generate multiple views, and maximizes the agreement between the global representations of these views. GCC [37] first samples multiple subgraphs for each graph by random walk, and then maximizes the agreement between the global representations of the subgraphs which belong to the same graph. Improving graph representation learning by contrastive regularization (Contrast-Reg) [38] proposes a lightweight local–local contrastive regularization term, and avoids the high scale of node representation norms and the high variance among them to improve the generalization performance. More recently, authors such as Hwang et al. [39] and Opolka et al. [40] studied different types of graphs, e.g., the heterogeneous graphs and the dynamic graphs. However, most of these previous C-UGRL methods have been developed without considering the relationship between the contrastive learning and the downstream task, resulting in a lack of empirical studies and theoretical understanding.

III. Motivation

Although we know that the contrastive learning has powerful ability on extracting embeddings, the extracted embeddings are feather applied on specific downstream tasks. Consequently, how to reduce the gap between the contrastive learning and the downstream task has emerged as an important issue when using this kinds of teleology in practical implementations. The key idea of contrastive learning based on MI maximization is to first formulate positive pairs (i.e., anchor embeddings and positive embeddings) and negative pairs (i.e., anchor embeddings and negative embeddings), and then ensure that positive pairs are closer than negative pairs in the embedding space. Specifically, given a feature map f_θ projecting the original feature matrix $\mathbf{X}$ to its embedding matrix $\mathbf{Z}$, where θ denotes the parameters of f_θ, the contrastive loss $\mathcal{L}^{\text{con}}(f_\theta)$ ensures the embedding of each anchor embedding (i.e., $\mathbf{z} \in \mathbf{Z}$) close to its positive embeddings (i.e., $\mathbf{z}^+ \in \mathbf{Z}^+$) while far away from its negative embeddings (i.e., $\mathbf{z}^- \in \mathbf{Z}^-$) [31], and thus resulting in the objective function $\mathcal{L}^{\text{con}}(f_\theta)$ as follows:

$$\mathcal{L}^{\text{con}}(f_\theta) = \mathop{\mathbb{E}}_{\mathbf{z},\mathbf{z}^+,\mathbf{z}^-}\left[\ell\left\{\mathbf{z}\cdot\left(\mathbf{z}^+\right)^T - \mathbf{z}\cdot\left(\mathbf{z}^-\right)^T\right\}\right] \tag{1}$$

where $\ell(\cdot)$ is a convex loss function, such as the hinge loss and logistic loss.

The purpose of representation learning is to serve downstream tasks, thus it is essential to analyze the connection between unsupervised representation learning and the downstream task (a classification task as an example in this article). Since neither the embeddings nor the samples are labeled in unsupervised learning, we first assume the existence of a set of latent classes $\mathcal{C}$ over the representation space $\mathcal{Z}$, and then let ρ be the probability distribution over $\mathcal{C}$. In the binary classification, we have $\mathcal{C} = \{c^+, c^-\}$, where c^+ and c^- represent the positive class and the negative class, respectively.

Denoting $\mathcal{L}^{\text{task}}(f_{\widehat{\theta}})$ as the loss of the downstream task (e.g., mean classifier) with optimized embeddings obtained by (1), $\mathcal{L}^{\text{con}}_{\neq}(f_\theta) = \mathbb{E}[\ell\{\mathbf{z}\cdot(\mathbf{z}^+ - \mathbf{z}^-)^T\}|c^+ \neq c^-]$ as the contrastive loss without class collision [31], $\tau = \mathbb{E}_{(c^+,c^-)\sim\rho^2}\mathbb{I}\{c^+ = c^-\}$ as the probability of anchor embeddings and negative embeddings belonging to the same latent class, and $s(f_\theta) = \mathbb{E}_{c\sim\rho}[\sqrt{\|\text{Var}(\mathbf{z})_c\|_2}\ \mathbb{E}_{\mathbf{z}\sim c}\|\mathbf{z}\|]$ as the intraclass deviation, Saunshi et al. [31] propose the following Lemma 1.

Lemma 1: With the probability $(1-\delta)$ over the training set, $\forall f \in \mathcal{F}$, the following inequality holds:

$$\mathcal{L}^{\text{task}}\left(f_{\widehat{\theta}}\right) \leq \mathcal{L}^{\text{con}}_{\neq}(f_\theta) + \gamma\, s(f_\theta) + (1+\gamma)\text{Gen}_M \tag{2}$$

where $\gamma = (1/1-\tau)-1$ and Gen_M is the generalization error controlled by the size M of the training set.

Equation (2) indicates that small value of the RHS of (2) enables low loss for the downstream task. Previous works (e.g., [11], [31], [34]) have made a lot of theoretical analysis on the relationship between contrastive learning and the downstream task, but few literature leverage the above theoretical analysis to design graph contrastive learning methods, which benefit the downstream tasks. In this article, we propose a novel framework in Fig. 1 to conduct unsupervised graph contrastive learning, whose outputted embeddings are suitable for the downstream tasks.

IV. Method

Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with nodes $v_i \in \mathcal{V}$ and edges $(v_i, v_j) \in \mathcal{E}$, we denote $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \in \mathbb{R}^{n\times d}$ as the node feature matrix (i.e., semantic information), where

each node has a d-dimensional representation. The structural information of $\mathbf{X}$ is represented by the graph $\mathbf{A} \in \mathbb{R}^{n\times n}$ where $a_{ij} = 1$ if $(v_i, v_j) \in \mathcal{E}$, otherwise $a_{ij} = 0$.

A. Semantic and Structural Information

In this article, we focus on the MI maximization for graph representation learning by considering both the semantic information (i.e., $\mathbf{X}$) and the structural information (i.e., $\mathbf{A}$) of the data based on the following motivations. First, as shown in Fig. 1, our consideration makes the definition of the related information in our framework simple as it does not need data augmentation. This also makes our method efficiently handle false negative embeddings and false connections (verified in Section IV-B). Second, the structural information is necessary for the GCN model to output discriminative representations as it is non-Euclidean data information containing the relationship between two nodes [42]. To meet the above scenarios, we define the embeddings of the anchors, the positives, and the negatives as follows by taking into account both the semantic information and the structural information for contrastive learning:

$$\begin{cases} \mathbf{Z} &= f_\theta(\mathbf{X}, \mathbf{I}), & \text{Anchor embedding matrix} \\ \mathbf{Z}^+ &= f_\theta(\mathbf{X}, \mathbf{A}), & \text{Positive embedding matrix} \\ \mathbf{Z}^- &= f_\theta(\mathbf{X}^-, \mathbf{I}), & \text{Negative embedding matrix} \end{cases} \tag{3}$$

where $f_\theta(\cdot)$ denotes the GNN encoder model (e.g., GCN [9] and GAT [43]) and $\mathbf{I}$ is an identity matrix. Similar to previous methods, negative samples $\mathbf{X}^-$ are drawn randomly from the input $\mathbf{X}$ to form negative embeddings, i.e., a row-wise random permutation in this article. It is noteworthy that the anchor embedding matrix $\mathbf{Z}$ only takes into account the semantic information $\mathbf{X}$ while the positive embedding matrix $\mathbf{Z}^+$ considers both the semantic information $\mathbf{X}$ and the structural information $\mathbf{A}$. Note that, it is difficult to directly maximize MI between the structural information and semantic information, because only using $\mathbf{A}$ is difficult to extract the embeddings of structural information. Therefore, as an alternative, we use the embeddings obtained by $f_\theta(\mathbf{X}, \mathbf{A})$ as the embeddings of structure information. On the contrary, previous methods need complicated pretext tasks to either generate positive embeddings or anchor embeddings such as $\mathcal{T}(\mathbf{X}, \mathbf{A})$ in augmentation methods (e.g., [15], [24], [44]) and $\mathcal{R}(\mathbf{Z})$ in cross-scale methods (e.g., [12], [35], [45]), where $\mathcal{T}(\cdot)$ is a transformation function and $\mathcal{R}(\cdot)$ is a readout function. More differences about the definitions of positive and negative embeddings between our method and others can be found in Table I, where $\{f_{\theta_0}, f_{\theta_1}, \ldots, f_{\theta_6}\}$ represent the encoders of models.

B. Constraints

According to Lemma 1, a contrastive learning framework fitting for downstream tasks (i.e., with good generalization ability) should make the value of $\mathcal{L}^{\text{con}}_{\neq}(f_{\widehat{\theta}}) + \gamma\, s(f_\theta) + (1+\gamma)\text{Gen}_M$ small. Actually, the value of Gen_M is related to the size M of the training set, but it is usually not considered to be improved in practice. Given the new definitions of positive and negative embeddings in Section IV-A, we design a new framework based on Theorem 1 to make the RHS of (2) be a small value.

TABLE I

DIFFERENT INFORMATION BETWEEN THE GRAPH CONTRASTIVE LEARNING METHODS

Methods	Representation pairs	Multual information
DGI [12]	$\mathbf{Z}^+ = f_{\theta_0}(\mathbf{X}, \mathbf{A})$ $\mathbf{h} = \mathcal{R}(\mathbf{Z}^+)$ $\mathbf{Z}^- = f_{\theta_0}(\mathcal{T}_0(\mathbf{X}, \mathbf{A}))$	$\mathbf{MI}\ (\mathbf{Z}^+, \mathbf{h})$
GIC[36]	$\mathbf{Z}^+ = f_{\theta_1}(\mathbf{X}, \mathbf{A})$ $\mathbf{Z}^- = f_{\theta_1}(\mathcal{T}_1(\mathbf{X}, \mathbf{A}))$ $\boldsymbol{\mu}_k = \frac{1}{\lvert\mathcal{C}_k\rvert}\sum_{v_i\in\mathcal{C}_k} \mathbf{z}_i^+$ $\mathbf{Z} = \sigma\left(\sum_{k=1}^K r_{ik}\boldsymbol{\mu}_k\right)$	$\mathbf{MI}\ (\mathbf{Z}, \mathbf{Z}^+)$
GRACE[13]	$\mathbf{Z} = f_{\theta_2}(\mathcal{T}_2(\mathbf{X}, \mathbf{A}))$ $\mathbf{Z}^+ = f_{\theta_2}(\mathcal{T}_3(\mathbf{X}, \mathbf{A}))$ $\mathbf{Z}^- = \left\{\mathbf{Z}_j \cup \mathbf{Z}_j^+ \lvert j \neq i\right\}$	$\mathbf{MI}\ (\mathbf{Z}^+, \mathbf{Z})$
GMI[41]	$\mathbf{Z}^+ = \mathbf{X}$ $\widetilde{\mathbf{Z}}^+ = \mathbf{A}$ $\mathbf{Z} = f_{\theta_3}(\mathbf{X}, \mathbf{A})$ $\widetilde{\mathbf{Z}} = \sigma\left(\mathbf{Z}\mathbf{Z}^T\right)$ $\mathbf{Z}^- = \mathbf{X}^-$ $\widetilde{\mathbf{Z}}^- = \mathbf{A}^-$	$\mathbf{MI}\ (\mathbf{Z}, \mathbf{Z}^+)$ $\mathbf{MI}\ (\widetilde{\mathbf{Z}}, \widetilde{\mathbf{Z}}^+)$
MVGRL[15]	$\mathbf{Z}^+ = f_{\theta_4}(\mathbf{X}, \mathbf{A})$ $\widetilde{\mathbf{Z}}^+ = f_{\theta_4}(\mathcal{T}_4(\mathbf{X}, \mathbf{A}))$ $\mathbf{h} = \mathcal{R}(\mathbf{Z}^+)$ $\widetilde{\mathbf{h}} = \mathcal{R}(\widetilde{\mathbf{Z}}^+)$ $\mathbf{Z}^- = f_{\theta_4}(\mathcal{T}_5(\mathbf{X}, \mathbf{A}))$ $\widetilde{\mathbf{Z}}^- = f_{\theta_4}(\mathcal{T}_6(\mathbf{X}, \mathbf{A}))$	$\mathbf{MI}\ (\mathbf{Z}^+, \widetilde{\mathbf{h}})$ $\mathbf{MI}\ (\widetilde{\mathbf{Z}}^+, \mathbf{h})$
Contrast-Reg[38]	$\mathbf{Z}^+ = f_{\theta_5}(\mathbf{X}, \mathbf{A})$ $\mathbf{h} = \mathcal{R}(\mathbf{Z}^+)$ $\mathbf{Z}^- = f_{\theta_5}(\mathcal{T}_7(\mathbf{X}, \mathbf{A}))$	$\mathbf{MI}\ (\mathbf{Z}^+, \mathbf{h})$
GRLC **(ours)**	$\mathbf{Z}^+ = f_{\theta_6}(\mathbf{X}, \mathbf{S})$ $\mathbf{Z} = f_{\theta_6}(\mathbf{X})$ $\mathbf{Z}^- = f_{\theta_6}(\mathbf{X}^-, \mathbf{I})$	$\mathbf{MI}\ (\mathbf{Z}, \mathbf{Z}^+)$

Theorem 1: The unsupervised graph contrastive learning framework meeting all following cases results in good generalization ability, i.e.,

Case 1: Small value of τ (i.e., class collision probability) reduces the upper bound of $\mathcal{L}^{\text{task}}(f_{\widehat{\theta}})$ in (2).

Case 2: Small value of the intraclass variance $\text{Var}(\mathbf{z})$ reduces the upper bound of $\mathcal{L}^{\text{task}}(f_{\widehat{\theta}})$ in (2).

Case 3: Small value of $\mathcal{L}^{\text{con}}_{\neq}$ reduces the upper bound of $\mathcal{L}^{\text{task}}(f_{\widehat{\theta}})$ in (2).

Case 4: Small value of the representation norm $\mathbb{E}\|f_\theta(\mathbf{x})\|$ reduces the upper bound of $\mathcal{L}^{\text{task}}(f_{\widehat{\theta}})$ in (2).

Proof: Given the probability of class collision τ and a convex loss function ℓ, we denote $\mu_c = \mathbb{E}_{\rho(\mathbf{z}|\mathbf{c})}[\mathbf{z}]$ be the mean vector of all embeddings that belong to the latent class c. Therefore, we can obtain the following inequation by Jensen's in-equality:

$$\begin{aligned} \mathcal{L}^{\text{con}}(f_\theta) &= \mathop{\mathbb{E}}_{c^+,c^-} \mathop{\mathbb{E}}_{\substack{(\mathbf{z},\mathbf{z}^+)\sim(c^+)^2 \\ \mathbf{z}^-\sim c^-}} \left[\ell\left(\mathbf{z}(\mathbf{z}^+ - \mathbf{z}^-)^T\right)\right] \\ &\geq^{(a)} \mathop{\mathbb{E}}_{c^+,c^-} \mathop{\mathbb{E}}_{\mathbf{z}\sim c^+} \left[\ell\left(\mathbf{z}(\mu_{c^+} - \mu_{c^-})^T\right)\right] \\ &=^{(b)} (1-\tau) \mathop{\mathbb{E}}_{c^+,c^-} \left[\mathcal{L}^{\text{task}}(f_\theta) \mid c^+ \neq c^-\right] + \tau \\ &= (1-\tau)\mathcal{L}^{\text{task}}(f_\theta) + \tau \end{aligned} \tag{4}$$

where (a) follows from the convexity of ℓ and Jensen's inequality by taking the expectation over $\mathbf{z}^+$,$\mathbf{z}^-$ inside the function, (b) follows by splitting the expectation into the cases $c^+ = c^-$ and $c^+ \neq c^-$, and $\ell(\mathbf{z}(\mu_{c^+} - \mu_{c^+})^T) = \ell(0) = 1$. We can partition negative embeddings into two categories, such as false negative embeddings (i.e., the anchor and its negative embeddings belonging to the same latent class) and real negative embeddings (i.e., the anchor and its negative embeddings belonging to the different latent classes). Different categories results in different losses, so the contrastive loss includes the loss induced by false negative embeddings [i.e., $\mathcal{L}^{\text{con}}_{=}(f_\theta)$] and the loss inducted by real negative embeddings [i.e., $\mathcal{L}^{\text{con}}_{\neq}(f_\theta)$]. As a result, (4) can be decomposed as

$$\mathcal{L}^{\text{con}}(f_\theta) = \tau \mathcal{L}^{\text{con}}_{=}(f_\theta) + (1-\tau)\mathcal{L}^{\text{con}}_{\neq}(f_\theta) \tag{5}$$

where $\mathcal{L}^{\text{con}}_{=}(f_\theta) = \mathbb{E}[\ell\{\mathbf{z}\cdot(\mathbf{z}^+ - \mathbf{z}^-)^T\}|c^+ = c^-]$ and $\mathcal{L}^{\text{con}}_{\neq}(f_\theta) = \mathbb{E}[\ell\{\mathbf{z}\cdot(\mathbf{z}^+ - \mathbf{z}^-)^T\}|c^+ \neq c^-]$.

Rearranging (4) and (5), we have

$$\mathcal{L}^{\text{task}}(f_\theta) \leq \mathcal{L}^{\text{con}}_{\neq}(f_\theta) + \frac{\tau}{1-\tau}\big(\mathcal{L}^{\text{con}}_{=}(f_\theta) - 1\big). \tag{6}$$

Similarly, with Jensen's in-equality, we can get $\mathcal{L}^{\text{con}}_{=}(f_\theta) > \ell(\mathbf{z}(\mu_{c^+} - \mu_{c^+})^T) = \ell(0) = 1$, which indicates that $(\mathcal{L}^{\text{con}}_{=}(f_\theta) - 1) > 0$ always holds. In this scenario, a small value of τ (i.e., a small probability of the class collision) results in low upper bound for downstream task, and thus *case 1* holds.

Following the proof in (5), we have

$$\begin{aligned}
\mathcal{L}^{\text{con}}_{\neq} &= \frac{1}{1-\tau}\big(\mathcal{L}^{\text{con}} - \tau\mathcal{L}^{\text{con}}_{=}\big) \\
&= \frac{1}{1-\tau}\Big(\mathcal{L}^{\text{con}} - \tau\mathbb{E}\Big[\ell\Big\{\mathbf{z}\cdot\big(\mathbf{z}^+ - \mathbf{z}^-\big)^T\Big\}|c^+ = c^-\Big]\Big) \\
&\leq \frac{1}{1-\tau}\big(\mathcal{L}^{\text{con}} - \tau\ell\big(\mathbf{z}(\mu_{c^+} - \mu_{c^+})^T\big)\big) \\
&= \frac{1}{1-\tau}(\mathcal{L}^{\text{con}} - \tau).
\end{aligned} \tag{7}$$

Combining (7) and Lemma 1, it is easy to see a small value of $\mathcal{L}^{\text{con}}$ reduces the upper bound of $\mathcal{L}^{\text{task}}$, and thus *case 3* holds.

Letting $\mathbf{v} = \mathbf{z}(\mathbf{z}^+ - \mathbf{z}^-)^T$ and $\ell(\mathbf{v}) = \max\{0, 1+\mathbf{v}\}$ be hinge loss, we have the following inequality:

$$\begin{aligned}
\mathcal{L}^{\text{con}}_{=} &= \mathbb{E}\big[(1+\mathbf{v})_+\big] \\
&\leq \mathbb{E}[\max\{1+\mathbf{v}, 1\}] \\
&\leq 1 + \mathbb{P}[\mathbf{v} \geq 0]\mathbb{E}[\mathbf{v}|\mathbf{v} \geq 0] \\
&\leq 1 + \mathbb{E}[|\mathbf{v}|] \\
&= 1 + \mathbb{E}_{\mathbf{z},\mathbf{z}^+,\mathbf{z}^-}\Big[\Big|\Big(\mathbf{z}\big(\mathbf{z}^+ - \mathbf{z}^-\big)^T\Big)\Big|\Big] \\
&\leq 1 + \mathbb{E}_{\mathbf{z}}\left[\|\mathbf{z}\|\sqrt{\mathbb{E}_{\mathbf{z}^+,\mathbf{z}^-}\left[\left(\frac{\mathbf{z}}{\|\mathbf{z}\|}(\mathbf{z}^+ - \mathbf{z}^-)^T\right)^2\right]}\right] \\
&\leq 1 + \sqrt{2}\mathbb{E}_{c\sim\rho}\Big[\sqrt{\|\text{Var}(\mathbf{z})_c\|_2}\ \mathbb{E}_{\mathbf{z}\sim c}\|\mathbf{z}\|\Big]
\end{aligned} \tag{8}$$

where $\text{Var}(\mathbf{z})_c$ is the intraclass variation and $\{\cdot\}_+ = \max\{\cdot, 0\}$. In this scenario, we have $\mathcal{L}^{\text{con}}_{=}(f_\theta) - 1 \leq ks(f_\theta) = k\mathbb{E}_{c\sim\rho}[\sqrt{\|\text{Var}(\mathbf{z})_c\|_2}\ \mathbb{E}_{\mathbf{z}\sim c}\|\mathbf{z}\|]$ where $k \in \mathbb{R}^+$. Combining (8) and (6), both *case 2* and *case 4* holds. ■

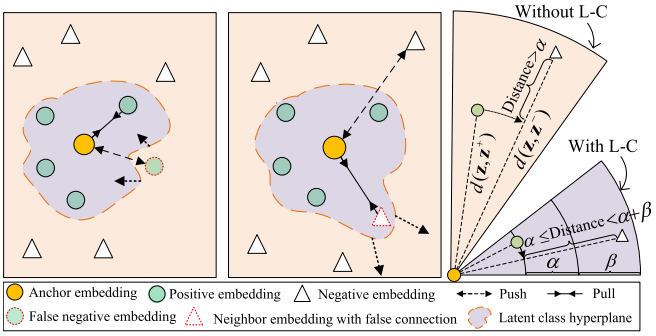

Fig. 2. Visualization of the cons of false negative embeddings (left) and false connection (middle), and the pros of our loss constraint ("L-C") (right). Either false negative embeddings or false connections stretch the latent class hyperplane.

In the following part of this section, we propose three constraints and a loss function to meet the above four cases for conducting contrastive learning.

1) Negative Embedding Constraint: As mentioned in *Case 1*, high probability of class collision makes the upper bound of the downstream task loss large, and thus the representation learning might be suboptimal. In supervised learning, since both negative embeddings and positive embeddings are selected based on the label information [46], the class hyperplane is constructed by positive embeddings (e.g., without false negative embeddings). In unsupervised representation learning, the latent class hyperplane is difficult to preserve due to the influence of false negative embedding. As shown in the left subfigure of Fig. 2, the latent class hyperplane can be stretched by the false negative embeddings, thereby restricting the discriminative and generalization ability of the obtained embeddings.

If both the class/cluster number and the sample size are large enough, the value of τ will be very small. As a result, the issue of the class collision can be ignored. However, in real applications, not all datasets contain many classes. To address this issue, in this article, we propose an adaptive solution to reduce the impact of class collisions by adaptively adjusting the weights of negative embeddings. More specifically, our solution automatically assigns small weights to punish false negative embeddings and large weights to encourage true negative embeddings. To do this, we first have the following definition.

Definition 1: If a negative embedding is similar to the anchor embedding based on a similarity measurement, they have high probability from the same latent class. This negative embedding is regard as a false negative embedding.

Based on Definition 1, we investigate adaptively updating the weights of negative embeddings, aiming at reducing the influence of false negative embeddings in the process of representation learning. Specifically, at each iteration of the optimization process, we randomly choose k independent and identically distributed negative samples $\{\mathbf{x}_1^-, \mathbf{x}_2^-, \ldots, \mathbf{x}_k^-\}$ for each anchor $\mathbf{x}_i$, followed by calculating the ℓ_2-norm distance between the anchor embedding $\mathbf{z}_i$ and the negative embedding $\mathbf{z}_k^-$, i.e., $d(\mathbf{z}_i, \mathbf{z}_k^-)$. Finally, the weight for each

negative embedding is defined as

$$\omega_i = \begin{cases} \left(\frac{d(\mathbf{z}_i, \mathbf{z}_k^-)}{\mathcal{D}_i}\right)^2, & d(\mathbf{z}_i, \mathbf{z}_k^-) < \mathcal{D}_i \\ 1, & d(\mathbf{z}_i, \mathbf{z}_k^-) \geq \mathcal{D}_i \end{cases} \tag{9}$$

where $\mathcal{D}_i = \mathbb{E}_{v_j \in \mathcal{N}_i}[d(\mathbf{z}_i, \mathbf{z}_j)]$ and $\mathcal{N}_i$ represents one-hop neighborhood set of the node v_i.

Proposition 1: Equation (9) assigns different weights to negative embeddings, which gradually alleviates the class collision in Case 1 of Theorem 1.

Proof: Each node v_i with its m-hop neighbors $\mathcal{N}_i^m$ forms a connected subgraph $\mathcal{G}_i$, where there exists a path for each pair of nodes. In the embedding space of all nodes in $\mathcal{G}_i$, we first define $\mathcal{D}_i^m$ as the expectation of the distance between the node v_i and its m-hop neighbors, and then obtain a subgraph area (i.e., a circle) $\mathcal{R}_i$ with the radius $\mathcal{D}_i^m$ and the center v_i. According to the principle of label propagation [47] and the assumption of graph smoothness [48], all embeddings in $\mathcal{G}_i$ gradually belong to the same latent class c_i with the increase of the optimization iterations. Given a negative embedding $\mathbf{z}_k^-$ belonging to a latent class c_k^-, we discuss the issue of class collision by two scenarios, i.e., $d(\mathbf{z}_i, \mathbf{z}_k^-) \geq \mathcal{D}_i^m$ and $d(\mathbf{z}_i, \mathbf{z}_k^-) < \mathcal{D}_i^m$.

If the scenario is true, i.e., $d(\mathbf{z}_i, \mathbf{z}_k^-) \geq \mathcal{D}_i^m$, $\mathbf{z}_k^-$ does not locate in the area $\mathcal{R}_i$, i.e., $c_i \neq c_k^-$. This case does not result in the issue of class collision. If the scenario is true, i.e., $d(\mathbf{z}_i, \mathbf{z}_k^-) < \mathcal{D}_i^m$, $\mathbf{z}_k^-$ locates in the area $\mathcal{R}_i$. As mentioned before, $\mathbf{z}_k^-$ tends to be categorized into the same latent class of $\mathcal{G}_i$, i.e., $c_i = c_k^-$. This exactly is the issue of the class collision. To solve this issue, our solution in (9) assigns the weight ω_k of the negative embedding $\mathbf{z}_k^-$ as $(d(\mathbf{z}_i, \mathbf{z}_k^-)/\mathcal{D}_i^m)^2$. In particular, if the inequality holds, i.e., $d(\mathbf{z}_i, \mathbf{z}_k^-) \ll \mathcal{D}_i^m$, we have $\omega_k \to 0$, which indicates the class collision is avoided. Therefore, (9) designs an adaptive method to gradually reduce the weights of negative embeddings for the scenario of $c_i = c_k^-$, and thus the class collisions is alleviated. ■

We obtain the observations from (9) as follows. If the negative embedding has the same latent class as the anchor embedding, i.e., class collision, it is a false negative embedding and is weighted by a small value or even zero [i.e., the upper row of (9)]. On the contrary, the true negative embedding with a low probability of class collision is weighted by 1. As a result, the optimization of the loss function is biased to the true negative embeddings. In particular, the same negative embedding has different weights in different iterations. That is, (9) adaptively adjusts the weights of negative embeddings. Furthermore, Fig. 3 illustrates that our negative embedding constraint reduces class collision, thereby meeting *Case 1* in Theorem 1.

2) Positive Embedding Constraint: The structural information $\mathbf{A}$ may contain incorrect edges, i.e., false connections between two nodes with different latent classes. In *Case 2*, large intraclass variance makes it difficult to separate the embeddings in downstream tasks. As shown in the middle subfigure of Fig. 2, one of the drawbacks of false connections is to enlarge the intraclass variance. To explain our solution

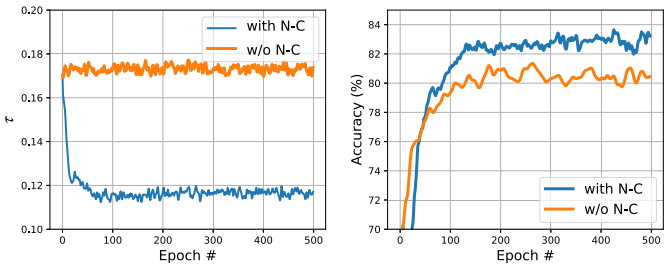

Fig. 3. Probability of class collision τ with the negative embedding constraint ("with N-C") and without the negative embedding constraint ("w/o N-C") on Dataset Cora (left). The accuracy curves of **GRLC** with the negative embedding constraint ("with N-C") and without the negative embedding constraint ("w/o N-C") on Dataset Cora (right).

for reducing the influence of false connections, we first have the following definition.

Definition 2: If an anchor embedding is not similar to the embedding of its connected node (i.e., neighborhood) based on a similarity measure, they have high probability of coming from different latent classes. Such a connection is called a false connection.

To reduce the impact of false connections in the graph $\mathbf{A}$, we employ the attention mechanism to assign different weights to each connection in the graph. More specifically, we first denote $e_{ij} = cos(\mathbf{z}_i, \mathbf{z}_j)$ as the weight of the edge between node i and node j. As a result, e_{ij} is a real value representing the similarity between the anchor embedding and the embedding of its neighborhood. We then combine e_{ij} with the original structural information $\mathbf{A}$ to have

$$s_{ij} = \frac{a_{ij} e_{ij}}{\sum_{j=1}^{n} a_{ij} e_{ij}}. \tag{10}$$

Proposition 2: Equation (10) reduces the influence of false connection, i.e., reducing the intraclass variance.

Proof: If the anchor embedding $\mathbf{z}_i$ has a false connection with the embedding $\mathbf{z}_k$, the positive embedding of $\mathbf{z}_i$ can be denoted as $\{f_{agg}(\mathbf{z}_i \cup \{e_{ij}\mathbf{z}_j, \forall v_j \in \mathcal{N}_i\} \cup e_{ik}\mathbf{z}_k)\}$, where f_{agg} is the aggregation operation of the GCN model. Meanwhile, $\mathbf{z}_k$ has a different latent class from $\mathbf{z}_i$. Thus, we have $d(\mathbf{z}_k, \mu_i) > d(\mathbf{z}_i, \mu_i)$ and $\mu_i = \mathbb{E}_{\rho(\mathbf{z}_i|\mathbf{c}_i)}[\mathbf{z}_i]$. It is noteworthy that e_{ik} equals e_{ij} $(v_j \in \mathcal{N}_i)$ if we do not consider false connection. Based on (10), the value of e_{ik} is less than the expectation value of e_{ij} $(v_j \in \mathcal{N}_i)$, so the value of $d(\mathbf{z}^+, \mu_i)$ reduces, compared to the case without considering the issue of false connection. Based on the definition of the variance, i.e., $\mathrm{Var}(\mathbf{z}_i^+)_{\mathbf{c}_i} = \mathbb{E}\|\mathbf{z}_i^+ - \mu_i\|^2 = \mathbb{E}_{\rho(\mathbf{z}_i^+|c_i)}[d(\mathbf{z}_i^+, \mu_i)]$ and $\mu_i = \mathbb{E}_{\rho(\mathbf{z}_i^+|\mathbf{c}_i)}[\mathbf{z}_i^+]$, (10) reduces the intraclass variance mentioned in Case 2 by reducing the impact of false connection. ■

Equation (10) defines the new structural information $\mathbf{S}$, which takes into account the confidence of the neighborhood. Hence, the positive embedding set defined in (3) is changed to $\widehat{\mathbf{Z}}^+ = f_\theta(\mathbf{X}, \mathbf{S})$. It is noteworthy that $\mathbf{S}$ is dynamically updated during the training process. Finally, our positive embedding constraint reduces the intraclass variance by adaptively updating the structural information.

3) Loss Functions: Based on *Case 3*, we need to optimize $f_\theta(\cdot)$ to explore rich contrastive relations among node

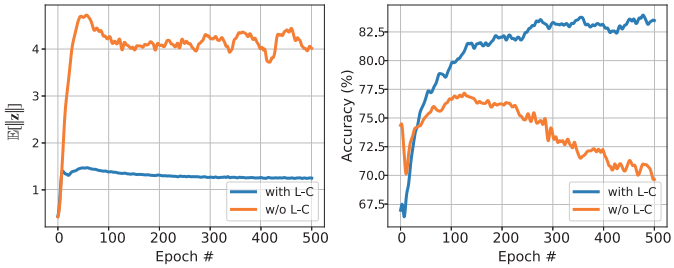

Fig. 4. Value of $\mathbb{E}[\|\mathbf{z}\|]$ of **GRLC** with the loss constraint ("with L-C") and without the loss constraint ("w/o L-C") on Dataset Cora (left). The accuracy curves of **GRLC** with the loss constraint ("with L-C") and without the loss constraint ("w/o L-C") on Dataset Cora (right).

embeddings by the contrastive loss. This is the fundamental part of contrastive learning. The popular contrastive loss functions include the logistic-based pairwise contrastive loss [6], the triplet loss [13], etc. Different loss functions can explore different contrastive relations. Interestingly, the triplet loss [49] is a special case of the logistic-based pairwise contrastive loss used in the Info noise contrastive estimation (InfoNCE) [6]. In this article, we focus on the triplet loss. Specifically, the triplet loss with respect to each embedding is formulated as

$$\mathcal{L}_{\text{triplet}} = \sum_{i=1}^{k} \left\{ d\left(\mathbf{z}, \widehat{\mathbf{z}}^{+}\right)^2 - d\left(\mathbf{z}, \mathbf{z}_i^{-}\right)^2 + \alpha \right\}_{-} \tag{11}$$

where $d(\cdot)$ is a similarity measurement (e.g., the ℓ_2-norm distance), α is a non-negative value to ensure "safe" distance between positive and negative embeddings, $\{\cdot\}_{-} = \min\{\cdot, 0\}$, and k is the number of negative embeddings. According to the inequality $\mathcal{L}_{\neq}^{\text{con}} \leq (1/1-\tau)(\mathcal{L}^{\text{con}} - \tau)$ (can be found in the proof of Theorem 1), minimizing (11) (i.e., minimizing $\mathcal{L}^{\text{con}}$) leads to a small value of $\mathcal{L}_{\neq}^{\text{con}}$.

4) Loss Constraint: Equation (11) encourages that the distance between $d(\mathbf{z}, \widehat{\mathbf{z}}^{+})$ and $d(\mathbf{z}, \mathbf{z}^{-})$ is larger than α. This can be easily achieved by increasing the value of $\mathbb{E}[\|\mathbf{z}\|]$ [38]. However, this easily results in both the positive embeddings and the negative embeddings far from the anchor embedding, as shown in the right subfigure of Fig. 2. Moreover, Fig. 4 shows a practical case on the real Dataset Cora.

According to *Case 4*, the small value of $\mathbb{E}[\|\mathbf{z}\|]$ makes contrastive learning with generalization ability. In this article, the triplet loss is revised to address the above issue, i.e., reducing the value of $\mathbb{E}[\|\mathbf{z}\|]$. Specifically, we investigate to add an upper bound for negative embeddings by

$$\alpha + d\left(\mathbf{z}, \widehat{\mathbf{z}}^{+}\right) < d\left(\mathbf{z}, \mathbf{z}^{-}\right) < d\left(\mathbf{z}, \widehat{\mathbf{z}}^{+}\right) + \alpha + \beta \tag{12}$$

where β is a non-negative value. As shown in the right subfigure of Fig. 2, the upper bound $(\alpha + \beta)$ guarantees the finite distance between negative embeddings and the anchor embedding, thus the distance between positive embeddings and anchor embeddings is also finite according to (11) (please see an example illustration on Dataset Cora in Fig. 4). As a result, the embeddings within the same latent class are indirectly constrained by the upper bound to avoid large value of $\mathbb{E}[\|\mathbf{z}\|]$.

C. Final Objective Function

Summing all triple losses for k negative embeddings, we obtain the loss for negative embeddings as follows:

$$\mathcal{L}_c = -\mathbb{E}\left[\frac{1}{k}\sum_{i=1}^{k}\left\{d\left(\mathbf{z}, \widehat{\mathbf{z}}^{+}\right)^2 - d\left(\mathbf{z}, \mathbf{z}_i^{-}\right)^2 + \alpha + \beta\right\}_{+}\right] \tag{13}$$

where $\{\cdot\}_{+} = \max\{\cdot, 0\}$. In particular, $d(\mathbf{z}, \widehat{\mathbf{z}}^{+})^2$ is set to stop the propagation of the gradient in (13) to prevent the increase of the intraclass variance. By integrating (11) with the loss constraints in (13), our final object function is formulated as

$$\mathcal{L} = \mathbb{E}\left[\frac{1}{k}\sum_{i=1}^{k}\left(w_i\left\{d\left(\mathbf{z}, \widehat{\mathbf{z}}^{+}\right)^2 - d\left(\mathbf{z}, \mathbf{z}_i^{-}\right)^2 + \alpha\right\}_{-} - \left\{d\left(\mathbf{z}, \widehat{\mathbf{z}}^{+}\right)^2 - d\left(\mathbf{z}, \mathbf{z}_i^{-}\right)^2 + \alpha + \beta\right\}_{+}\right)\right]. \tag{14}$$

Compared to (11), (14) adds 1) one more variable w_i defined in (9) for the embedding of each sample, aiming at adjusting the weight to reduce the impact of the false negative embeddings and 2) an upper bound loss in (13) to avoid a large value of $\mathbb{E}[\|\mathbf{z}\|]$. We list the whole training process of our proposed **GRLC** method in Algorithm 1.

Algorithm 1 Pseudo Code of Our Proposed **GRLC**

Input: Feature matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$,
adjacency matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$,
Output: Graph neural network encoder: $f_\theta()$

```
Initialize parameters; S = A
for t ← 1, 2, ··· , T do
    Obtain anchor embeddings Z ← f_θ(X, I)
    Obtain positive embeddings Z̃⁺ ← f_θ(X, S)
    Obtain negative embeddings Z⁻ ← f_θ(X⁻, I)
    w_{i∈n} ← (9)   ▷ Adjust weight for each negative embedding
    Loss ← (14)   ▷ With loss constrain
    Updating θ using optimizers, e.g., Adam
    S ← (10)   ▷ Adjust structural information for t+1 epoch
end for
```

V. Experiments

A. Experimental Setting

1) Datasets: The datasets included citation network datasets (i.e., Cora, Citeseer, PubMed, DBLP, and CoraFull) [51], [52], [53], page network datasets (i.e., Wiki-CS and Croco) [54], [55], Amazon sale dataset (i.e., Photo) [56], and three large-scale datasets (i.e., Ogbn-arxiv, Ogbn-mag, and Ogbn-products) [57]. For all public datasets, we used their public splitting settings in our experiments. Specifically, we report statistics of real-world datasets in Table VI of the Appendix.

2) Comparison Methods: The comparison methods included one traditional unsupervised graph learning method (i.e., DeepWalk [50]), two semi-supervised learning methods (i.e., GCN [9] and graph attention network (GAT) [43]), and six C-UGRL methods (i.e., DGI [12], GIC [36],

TABLE II
CLASSIFICATION ACCURACY (MEAN ± STANDARD DEVIATION) OF ALL METHODS ON EIGHT DATASETS

Methods	Cora	Citeseer	PubMed	Photo	Wiki-CS	DBLP	Croco	CoraFull
Raw Feature	55.1±0.6	57.8±0.5	69.1±0.8	78.5±0.5	72.0±0.9	71.6±0.6	41.7±0.4	43.6±0.7
DeepWalk [50]	67.2±0.5	63.2±0.6	75.3±0.6	89.4±0.5	74.4±0.8	76.0±0.7	42.5±0.7	53.2±0.5
GCN [9]	81.5±0.9	70.3±0.8	79.0±0.6	91.6±0.6	74.0±0.7	77.8±0.5	52.6±0.8	59.4±0.6
GAT [43]	83.0±0.7	72.5±0.7	79.0±0.3	91.8±0.6	77.6±0.6	78.2±1.5	53.3±1.0	58.6±0.5
DGI [12]	82.3±0.8	71.5±0.7	76.8±0.6	89.4±1.1	74.8±0.7	83.1±0.5	53.1±0.7	55.1±0.6
GIC [36]	81.7±0.9	71.9±1.4	77.3±1.1	91.6±0.9	75.9±0.6	81.9±0.8	56.8±0.6	58.2±0.7
GRACE [13]	83.1±0.7	72.1±0.6	79.6±1.0	90.1±0.5	75.3±0.7	**84.2**±0.6	58.3±0.4	54.0±0.6
GMI [41]	83.0±0.6	72.4±0.7	79.9±0.6	90.7±0.6	74.8±0.7	83.9±0.8	54.3±0.9	54.6±0.8
MVGRL [15]	82.9±0.8	72.6±0.7	80.1±0.7	91.7±0.6	76.3±1.1	79.5±0.8	57.9±0.6	58.8±0.7
Contrast-Reg [38]	82.7±0.6	**72.9**±0.7	80.1±0.6	91.5±0.7	77.0±0.6	83.6±0.8	58.4±0.7	58.9±0.6
GRLC (ours)	**83.5**±0.5	72.6±0.6	**82.1**±0.4	**92.3**±0.5	**77.9**±0.5	**84.2**±0.6	**59.5**±0.7	**59.4**±0.6

GRACE [13], GMI [41], MVGRL [15], and Contrast-Reg [38]). Raw features were directly used to conduct the downstream tasks.

3) Setting-Up: We conduct experiments on a server with Tesla V100 (32 GB memory each) and Intel(R) Xeon(R) E5-2678 CPU. All methods are implemented with the PyTorch (vision 1.9) framework. For all experiments, we repeated the experiments ten times with random seeds for all methods and then reported the average results and the corresponding standard deviation. We obtained the codes of all comparison methods from the authors or online and used their default parameter settings.

We evaluated the performance by classification accuracy for node classification task, by the area under ROC curve (AUC) score and average precision (AP) score for link predictions, and accuracy (Acc), normalized MI (NMI), and average rand index (ARI) for clustering task.

4) Implementation Details: In **GRLC**, we applied the rectified linear unit (ReLU) function [58] as a nonlinear activation for each layer and conducted the row normalization on input features. We employ one MLP layer followed by one GCN layer as the encoder across all datasets. The hyper-parameter α is set around 0.7 and β around 0.2. If the values of α are too small, **GRLC** obtains poor results as a small margin between positive and negative pairs making them difficult be distinguished. In contrast, large values of β easily lead to the upper bound in (14) becoming useless. Moreover, a dropout function is applied behind each layer. Additionally, all parameters were initialized with Glorot initialization [59] and optimized with Adam optimizer. To evaluate node representation, we follow the standard linear evaluation protocol introduced by Velickovic et al. [12]. Specifically, after we obtain the embeddings (i.e., $\mathbf{Z}$) of nodes in the graph, we apply the standard linear evaluation protocol on a small portion (the label ratio of all datasets can be found in Section VI-A) of labeled data (i.e., labeled nodes in the graph) using cross-entropy loss for a particular classification task. For the cluster task, we apply soft k-means clustering after generating the embeddings of nodes in the graph by **GRLC**. Link prediction is a task to estimate the probability of edges between nodes in a graph, we inferred the real links by obtained embeddings.

TABLE III
CLASSIFICATION ACCURACY (MEAN ± STANDARD DEVIATION) OF ALL METHODS ON THREE LARGE-SCALE DATASETS (I.E., OGBN-ARXIV, OGBN-MAG, AND OGBN-PRODUCTS)

Methods	ogbn-arxiv	ogbn-mag	ogbn-products
Raw Feature	56.3±0.3	22.1±0.3	59.7±0.2
DeepWalk [50]	63.6±0.4	25.6±0.3	73.2±0.2
GCN [9]	70.4±0.3	30.1±0.3	81.6±0.4
GAT[43]	70.6±0.3	30.5±0.3	82.4±0.4
DGI [12]	67.9±0.4	30.6±0.3	77.9±0.4
GIC[36]	67.8±0.4	29.8±0.2	75.8±0.4
GRACE [13]	67.4±0.4	31.1±0.3	77.4±0.4
GMI [41]	67.1±0.2	27.2±0.1	76.8±0.4
MVRLG [15]	68.1±0.1	30.4±0.4	78.1±0.4
Contrast-Reg [38]	68.2±0.1	30.9±0.4	78.3±0.4
GRLC (ours)	**68.6**±0.2	**31.6**±0.2	**82.5**±0.2

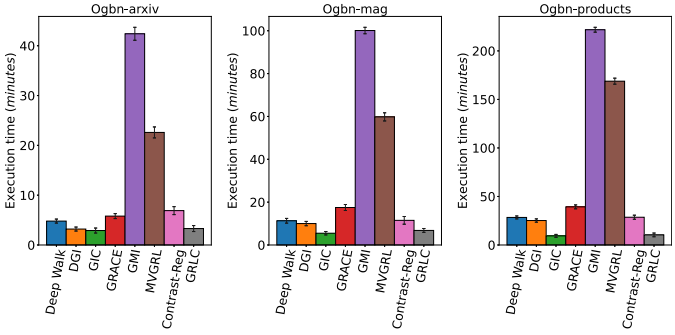

Fig. 5. Execution time (minutes) of all unsupervised graph representation methods on three large-scale datasets.

In Appendix Section VI-A, we describe hyperparameter and architectural details.

B. Results and Analysis

1) Node Classification: Tables II and III summarize the classification accuracy of all methods on eleven real graph-structure datasets, including three large-scale graph-structure datasets, i.e., Ogbn-arxiv, Ogbn-mag, and Ogbn-products.

From Table II, we have the following observations. First, our method achieves the best performance compared to the self-supervised methods, followed by Contrast-Reg, MVGRL,

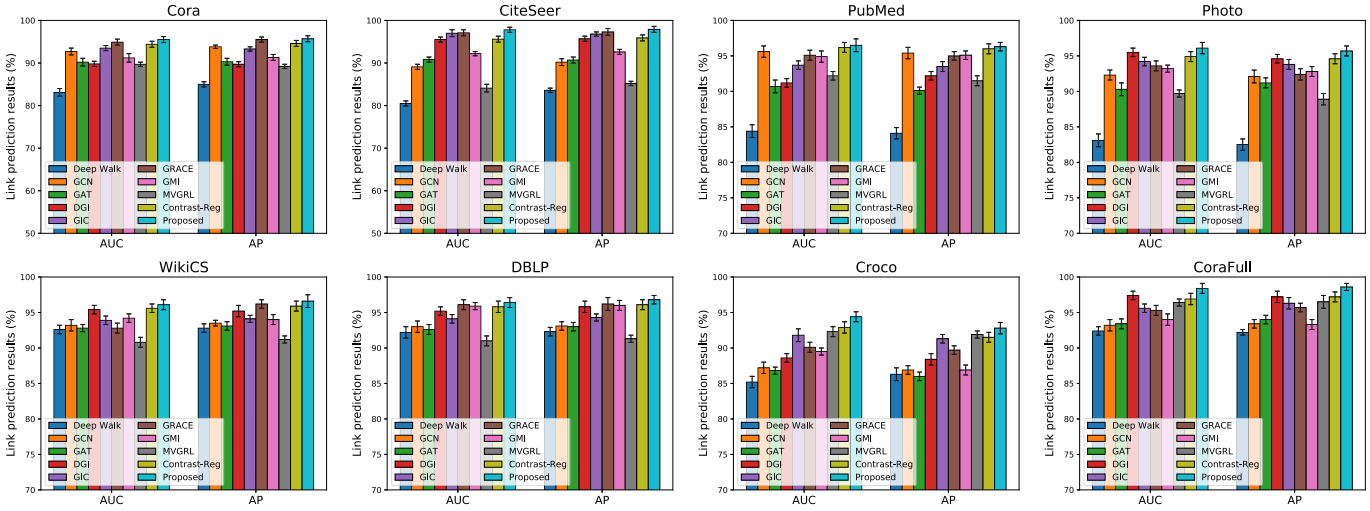

Fig. 6. Link prediction performance [i.e., AUC (%) and AP (%)] of all methods on eight datasets.

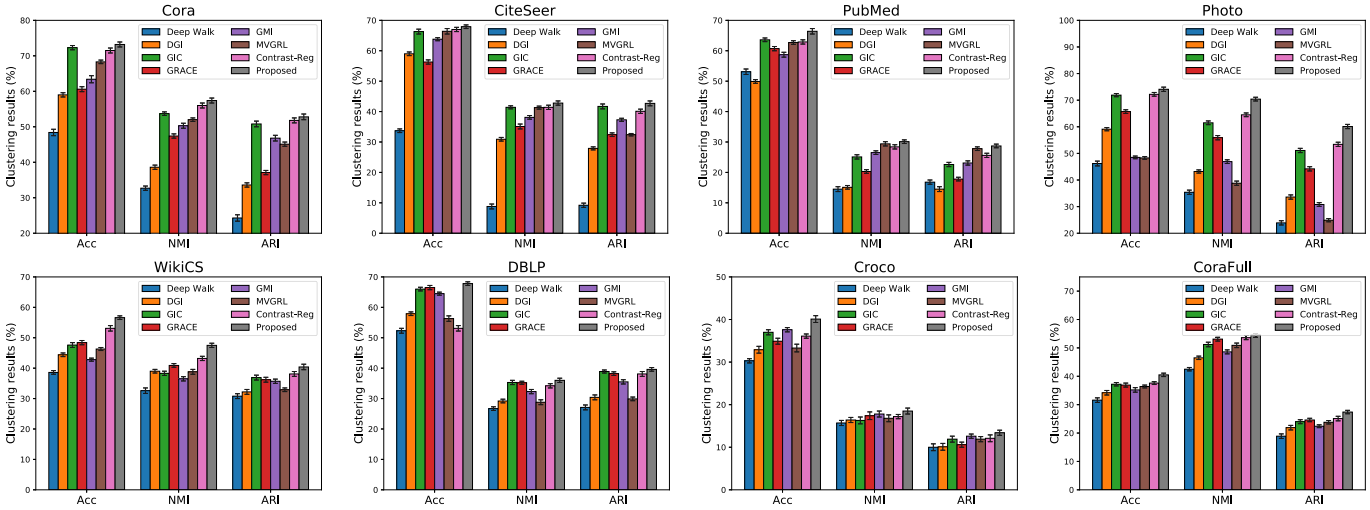

Fig. 7. Clustering performance [i.e., ACC (%), NMI (%), and ARI (%)] of all unsupervised methods on eight datasets.

GRACE, GIC, GMI, and DGI. For example, our method on average improves by 4.8%, and 1.1%, respectively, compared to the baseline self-supervised comparison method DGI and the best self-supervised comparison method Contrast-Reg. Moreover, our method consistently outperforms two baseline semi-supervised methods. For example, our method on average improves by 4.6% and 3.2%, respectively, compared to GCN and GAT, on eight datasets.

From Table III, we can find that our method achieves the superior performance on three large-scale datasets. For example, compared to the self-supervised methods (i.e., DGI, GIC, GRACE, GMI, MVGRL, and Contrast-Reg), **GRLC** improves by 1.7% on average, and even achieves similar performance to the supervised methods (i.e., GCN and GAT). Moreover, we test the time costs of all unsupervised graph representation methods on three large-scale datasets (i.e., ogbn-arxiv, ogbn-mag, and ogbn-products) and report Execution time (*minutes*) of all methods in Fig. 5. Obviously, the time costs of **GRLC** is low. This indicates that **GRLC** has good scalability.

2) Link Prediction: Fig. 6 illustrates the results of all methods on eight datasets. It can be seen that our **GRLC** achieves better performance than the best comparison Contrast-Reg. In particular, our method achieves about 1.1%–16.3% and 1.2%–16.9% improvement in terms of AUC and AP, compared to all comparison methods.

3) Clustering: Fig. 7 shows the clustering performance of all unsupervised methods on eight datasets. Obviously, our proposed **GRLC** beats all comparison methods on all datasets, in terms of ACC, NMI, and ARI, which suggest that **GRLC** can generate discriminative representations.

4) Results Analysis: In summary, **GRLC** achieves superior performance over all comparison methods on different downstream tasks, including node classification, link prediction, and clustering. In addition, our method has statistically significant difference from every comparison method because the p-values of all cases are < 0.035 on the paired-sample t-tests at the 95% significance level on downstream tasks. The reasons can be summarized as follows. **GRLC** uses the false negative embedding constraint to reduce the probability of class collisions, and the positive embedding constraint to reduce the influence of false connections, and meanwhile investigates an upper bound to reduce the representation norm. In this way,

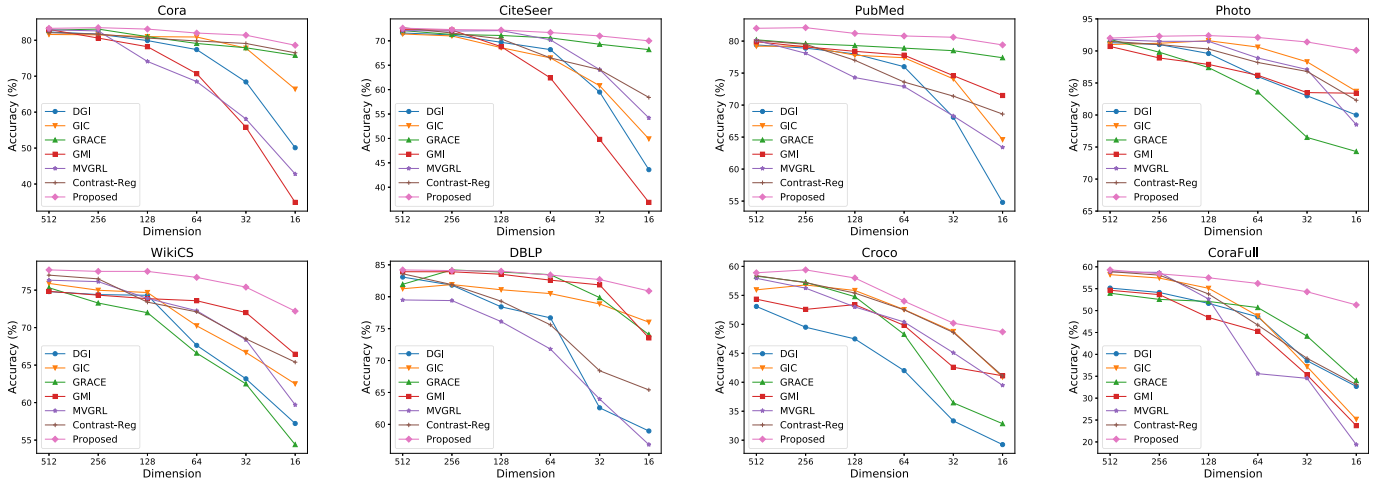

Fig. 8. Accuracy of all methods with different numbers of embedding dimensions on eight datasets. The performances of **GRLC** were robust within a certain range (i.e., the number of embedding dimensions $>$ 128).

TABLE IV
PERFORMANCE OF **GRLC** WITH DIFFERENT GNN ARCHITECTURE

GNN architectures	Cora	Citeseer	PubMed	Photo	Wiki-CS	DBLP	Croco	CoraFull
GCN-Conv [9]	$83.5_{\pm 0.5}$	$72.6_{\pm 0.6}$	$82.1_{\pm 0.4}$	$92.3_{\pm 0.5}$	$77.9_{\pm 0.5}$	$84.2_{\pm 0.6}$	$59.5_{\pm 0.7}$	$59.4_{\pm 0.6}$
EDGA-Conv [60]	$83.2_{\pm 0.5}$	$72.1_{\pm 0.5}$	$81.7_{\pm 0.6}$	$92.0_{\pm 0.4}$	$75.2_{\pm 0.7}$	$84.4_{\pm 0.5}$	$59.0_{\pm 0.6}$	$59.7_{\pm 0.7}$
GCNII-Conv [61]	$82.6_{\pm 0.7}$	$71.2_{\pm 0.8}$	$80.9_{\pm 0.6}$	$91.7_{\pm 0.5}$	$75.6_{\pm 0.6}$	$83.2_{\pm 0.6}$	$55.8_{\pm 0.8}$	$57.6_{\pm 0.7}$
TAG-Conv [62] *(Number of hops = 1)*	$83.7_{\pm 0.4}$	$72.5_{\pm 0.4}$	$82.1_{\pm 0.5}$	$92.2_{\pm 0.6}$	$78.3_{\pm 0.5}$	$84.8_{\pm 0.2}$	$59.0_{\pm 0.6}$	$59.3_{\pm 0.6}$
TAG-Conv [64] *(Number of hops = 2)*	$83.5_{\pm 0.7}$	$72.3_{\pm 0.9}$	$82.7_{\pm 0.4}$	$92.4_{\pm 0.5}$	$77.1_{\pm 0.6}$	$84.3_{\pm 0.4}$	$59.1_{\pm 0.6}$	$59.2_{\pm 0.6}$
TAG-Conv [64] *(Number of hops = 3)*	$81.1_{\pm 1.0}$	$70.3_{\pm 1.3}$	$80.4_{\pm 0.8}$	$91.4_{\pm 0.7}$	$74.9_{\pm 0.6}$	$82.7_{\pm 0.7}$	$54.0_{\pm 0.4}$	$57.6_{\pm 0.6}$
SAGE-Conv [63] *(Aggregator = 'mean')*	$82.0_{\pm 0.7}$	$71.1_{\pm 0.6}$	$83.4_{\pm 0.5}$	$91.3_{\pm 0.7}$	$77.4_{\pm 0.5}$	$82.7_{\pm 0.6}$	$57.6_{\pm 0.9}$	$58.4_{\pm 0.5}$
SAGE-Conv [63] *(Aggregator = 'gcn')*	$83.2_{\pm 0.7}$	$72.4_{\pm 0.7}$	$82.0_{\pm 0.6}$	$91.1_{\pm 0.5}$	$78.6_{\pm 0.7}$	$83.8_{\pm 0.6}$	$58.2_{\pm 0.7}$	$59.0_{\pm 0.6}$
SAGE-Conv [63] *(Aggregator = 'pool')*	$81.5_{\pm 0.8}$	$70.6_{\pm 0.7}$	$81.1_{\pm 0.7}$	$92.1_{\pm 0.8}$	$77.3_{\pm 0.9}$	$83.4_{\pm 0.7}$	$58.7_{\pm 0.7}$	$59.1_{\pm 0.9}$
GAT-Conv [43] *(Number of heads = 2)*	$84.1_{\pm 0.5}$	$72.3_{\pm 0.6}$	$82.8_{\pm 0.3}$	$92.9_{\pm 0.5}$	$78.4_{\pm 0.4}$	$82.6_{\pm 0.7}$	$56.0_{\pm 0.9}$	$60.4_{\pm 0.8}$
GAT-Conv [43] *(Number of heads = 4)*	$84.3_{\pm 0.7}$	$72.9_{\pm 0.9}$	$82.5_{\pm 0.6}$	$93.1_{\pm 0.8}$	$78.2_{\pm 0.5}$	$84.6_{\pm 0.7}$	$57.7_{\pm 0.6}$	$60.8_{\pm 0.6}$
GAT-Conv [43] *(Number of heads = 8)*	$84.5_{\pm 0.7}$	$73.0_{\pm 0.7}$	$82.4_{\pm 0.4}$	$93.4_{\pm 0.6}$	$77.9_{\pm 0.6}$	$83.7_{\pm 0.8}$	$57.1_{\pm 0.6}$	$60.1_{\pm 0.8}$

GRLC is able to reduce the gap between contrastive learning and downstream tasks to achieve significant generalization ability for UGRL.

5) Effectiveness of Low-Dimensional Representations: We further investigated the robustness of our method from another perspective, i.e., the effectiveness of low-dimensional representations. Obviously, low-dimensional representations with low complexity are preferred in real applications. To do this, we conducted experiments to generate deep features with different numbers of embedding dimensions (i.e., [512, 256, 128, 64, 32, 16]) for all methods and reported the classification accuracy in Fig. 8. Obviously, our method achieves the best results even for the small number of representation dimensions. In particular, the classification performance of our method varies a little with the decreasing of the representation dimension. On the contrary, the classification accuracy of DGI drastically decreases while the number of embedding dimensions is less than 128.

*6) Effectiveness of **GRLC** With Different GNN Architectures:* As mentioned in Section IV, the encoder model in **GRLC** can be different GNN architectures (e.g., GCN [9] and GAT [43]). To test the influence of **GRLC** with different GNN architectures, we conducted experiments to test performance of **GRLC** with different GNN architectures, such as i.e., **EDGA-Conv** [60], **GCNII-Conv** [61], **TAG-Conv** [62], **SAGE-Conv** [63], and **GAT-Conv** [43]. Based on Table IV, the performance of our method cloud be further improved with a more powerful graph learning encoder (e.g., **GAT-Conv** [43]), which is consistent with assumptions that a stronger encoder is likely to produce better performance. However, the improvement does not have statistically significant difference.

TABLE V

CLASSIFICATION ACCURACY (MEAN ± STANDARD DEVIATION) ON DIFFERENT COMBINATIONS OF CONSTRAINTS [NEGATIVE EMBEDDING CONSTRAINT (N-C), POSITIVE EMBEDDING CONSTRAINT (P-C), AND LOSS CONSTRAINT (L-C)] IN OUR METHOD ON EIGHT DATASETS. √ INDICATES HAVING THE CONSTRAINT

Cora	Citeseer	PubMed	Photo	Wiki-CS	DBLP	Croco	CoraFull	N-C	P-C	L-C
69.1±0.7	68.7±1.0	62.8±2.7	91.5±0.8	74.4±0.6	82.0±0.7	52.1±0.9	53.5±1.2	-	-	-
72.5±0.7	68.8±0.9	70.7±1.3	91.6±0.7	74.1±0.8	82.4±0.6	54.8±0.7	53.7±1.4	√	-	-
70.5±0.6	69.0±1.0	77.4±0.6	91.3±0.5	75.4±0.5	82.1±0.9	56.0±0.4	53.1±1.1	-	√	-
78.6±0.8	68.2±0.9	81.6±0.7	91.8±0.6	75.3±0.6	83.5±0.5	55.5±0.7	58.5±1.2	-	-	√
74.6±0.5	69.2±0.6	77.8±0.6	91.6±0.5	75.7±0.7	82.9±0.7	55.0±0.9	56.1±1.0	√	√	-
82.8±0.4	72.1±0.5	81.7±0.7	91.9±0.6	76.7±0.5	83.4±0.4	55.9±0.5	58.7±0.8	√	-	√
80.6±0.5	71.5±0.5	81.2±0.4	92.2±0.6	77.4±0.6	83.8±0.8	57.3±0.7	59.1±0.7	-	√	√
83.5±0.5	**72.6**±0.6	**82.1**±0.4	**92.3**±0.5	**77.9**±0.5	**84.2**±0.6	**59.5**±0.7	**59.4**±0.6	√	√	√

C. Ablation Study

1) Effectiveness of Constraints on Classification Accuracy: This section we test the effectiveness of our proposed constraints to reduce the gaps between self-supervised graph representation learning and downstream tasks. Specifically, our proposed method has three constraints, such as negative embedding constraint (N-C) to meet *Case 1*, positive embedding constraint (P-C) to meet *Case 2*, and loss constraint (L-C) to meet *Case 4*. We investigate the effectiveness of individual constraint by reporting the accuracy results in Table V, where the results in the first row did not consider any constraints and the results in the last row considered all three constraints. First, regarding the results in the first four rows, the loss constraint achieves the best improvement, followed by the positive embedding constraint and the negative embedding constraint. For example, the negative embedding constraint, the positive embedding constraint, and the loss constraint improve by 2.92%, 4.21%, and 7.77%, respectively, on average compared to the results in the first row, on all datasets. This indicates that each constraint in our method does work and *Case 4* is the most important case compared to *Case 1* and *Case 2* in our experiments. Second, the more constraints our method contains, the better classification performance of our method. As a result, our method considering all three constraints (i.e., in the last row) achieves the best results. For example, our method increases by 1.48%, 3.35%, and 11.3%, compared to the best results with two constraints, the best results with only one constraint, and the result without constraints, indicating that each constraint has a role to play.

2) Effectiveness of Constraints on $\mathcal{L}^{task}(f_{\hat{\theta}})$: Table V verified the feasibility of three constraints in terms of classification accuracy, we continue to verify them in terms of the value of $\mathcal{L}^{\text{task}}(f_{\hat{\theta}})$ (e.g., mean classifier) on Dataset Cora, shown in Fig. 9. In our experiments, we normalize the value of $\mathcal{L}^{\text{task}}(f_{\hat{\theta}})$ into $[0, 1]$. It is clear that the method without the three constraints result in large value of $\mathcal{L}^{\text{task}}(f_{\hat{\theta}})$ or even model collapse. Hence, the necessary of our proposed constraints is further verified.

3) Effectiveness of the Number of Negative Samples: In this section, we analyze the effectiveness of different numbers of negative samples on our method. For this purpose, we set the range of the number of negative samples k as [3, 5, 10, 15, 20, 30]. As shown in Fig. 10, the classification accuracy of our method increases with the increasing values of k, i.e., from $k = 3$ to $k = 10$, and increasing the number of negative samples beyond a threshold can hurt the performance, i.e., from $k = 15$ to $k = 30$. The reason might be that a small k value cannot fully exert the ability of contrastive learning, and a large value of k will cause frequent class collisions and prevent contrastive learning from learning representations. This shows that the performance of our method steadily improves as more negative examples are used, and achieves the best when the number of negative samples is greater than a certain threshold (e.g., $k = 5$). In other words, our method requires only a small number (i.e., around 5) of negative samples, which greatly saves computation and memory costs.

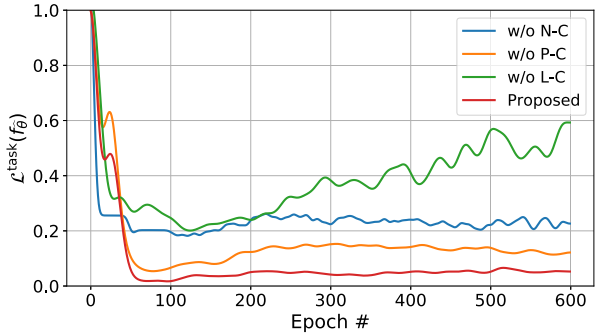

Fig. 9. Loss curves of $\mathcal{L}^{\text{task}}(f_{\hat{\theta}})$ under different combinations of constraints on Dataset Cora. Note that the x-axis represents update epochs of contrastive learning and y-axis represents the $\mathcal{L}^{\text{task}}(f_{\hat{\theta}})$ using the optimized embeddings $\mathbf{Z}$ obtained by embedding function under current update epoch (i.e., **NOT** the training update epochs of the downstream task).

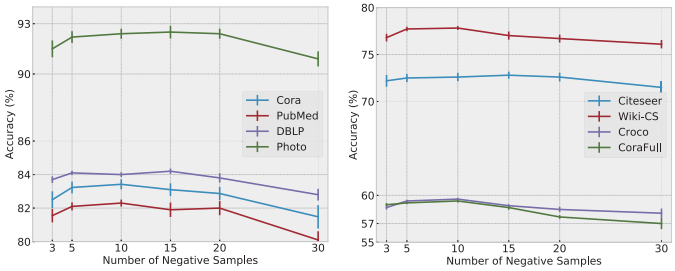

Fig. 10. Classification accuracy (mean ± standard deviation) of our method with different number of negative samples on node classification benchmarks.

TABLE VI
STATISTICS OF THE DATASETS

Dataset	Nodes	Edges	Features	Class	Split	Training Nodes	Validation Nodes	Testing Nodes
Cora	2708	5429	1433	7	fixed	140	500	1000
CiteSeer	3327	4732	3703	6	fixed	120	500	1000
PubMed	19717	44338	500	3	fixed	60	500	1000
Photo	7650	119081	745	8	random	762	762	6888
WikiCS	11701	297110	300	10	fixed	580	1769	5847
DBLP	17716	52867	1639	4	random	1770	1770	15946
Croco	11631	180020	500	6	random	1160	1160	10471
CoraFull	19793	63421	500	70	random	1953	1953	17840
Ogbn-arxiv	169343	1166243	128	40	random	16934	16934	135475
Ogbn-mag	736389	10792672	128	349	random	73638	73638	589113
Ogbn-products	2449029	61859140	100	47	random	244902	244902	1959225

TABLE VII
SETTINGS FOR **GRLC**

	Cora	CiteSeer	PubMed	Photo	WikiCS	DBLP	Croco	CoraFull	Ogbn-arxiv	Ogbn-mag	Ogbn-products
Dim_Embedding	256	256	128	128	512	512	256	256	128	128	128
Dim_Hidden	512	512	512	512	512	512	512	512	512	512	512
α	0.8	0.7	0.4	0.9	0.9	0.7	0.9	0.9	0.9	0.9	0.9
β	0.2	0.4	0.1	0.2	0.1	0.1	0.2	0.6	0.1	0.1	0.1
Epochs	500	500	5000	2000	5000	400	2000	4000	100	100	100
Num_Negatives	10	10	5	5	5	5	5	5	1	1	1
Lr	0.005	0.005	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Wd	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.00005	0.00005	0.00005
Dropout	0.2	0.2	0.2	0.1	0.1	0.1	0.1	0.1	0.2	0.2	0.2

VI. CONCLUSION

In this work, we proposed a new C-UGRL method, referred as **GRLC**, which maximizes the MI between the semantic information and the structural information to output discriminative embeddings. To tackle the realistic but ignored issue of gaps between contrastive learning and downstream tasks, we proposed three constraints and verified the effectiveness of each constraint, which is consistent with our theoretical implications. We demonstrated that **GRLC** improved the generalization ability of representation learning via extensive experiments on eleven public datasets. Note that MI maximization requires structural information. Incorporating structure learning into **GRLC** to extend its application without providing structural information is an interesting direction for our future works.

APPENDIX
GRLC: GRAPH REPRESENTATION LEARNING WITH CONSTRAINTS

Roadmap of Appendix: The Appendix is organized as follows. We provide the details of all datasets in Section VI-A. The detailed model architectures are shown in Section VI-B.

A. Dataset Details

See Table VI.

B. Implementation Details

In **GRLC**, we employ one MLP layer followed by one GCN layer as the encoder across all datasets, which are implemented with PyTorch framework. Specifically, the number of embedding dimensions after the MLP layer is represented by "Dim_Hidden" and the number of representation dimensions of the GCN layer is represented by "Dim_Embedding" We applied the ReLU function [58] as a nonlinear activation for each layer and conducted the row normalization on input features. Moreover, a dropout function is applied behind each layer, and the dropout rate is denoted as "Dropout" Additionally, all parameters were initialized with Glorot initialization [59] and optimized with Adam optimizer. For the optimizer, we use "Lr" to denote the initial learning rate and "W" to denote the weight decay. The number of negative samples for each anchor point is denoted as "Num_Negatives" To evaluate node representation, we follow the standard linear evaluation protocol introduced by Velickovic et al. [12]. Table VII describes hyperparameter and architectural details.

REFERENCES

[1] B. Zhang, Q. Qiang, F. Wang, and F. Nie, "Flexible multi-view unsupervised graph embedding," *IEEE Trans. Image Process.*, vol. 30, pp. 4143–4156, 2021.
[2] S. Pan, R. Hu, G. Long, J. Jiang, L. Yao, and C. Zhang, "Adversarially regularized graph autoencoder for graph embedding," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 2609–2615.
[3] J. Gan et al., "Multigraph fusion for dynamic graph convolutional network," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, May 16, 2022, doi: 10.1109/TNNLS.2022.3172588.
[4] C. Tang et al., "Learning a joint affinity graph for multiview subspace clustering," *IEEE Trans. Multimedia*, vol. 21, no. 7, pp. 1724–1736, Jul. 2019.
[5] M. Jin, Y. Zheng, Y.-F. Li, C. Gong, C. Zhou, and S. Pan, "Multi-scale contrastive Siamese networks for self-supervised graph representation learning," in *Proc. 13th Int. Joint Conf. Artif. Intell.*, Aug. 2021, pp. 1477–1483.

[6] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv:1807.03748*.

[7] Y. Liu, Z. Li, S. Pan, C. Gong, C. Zhou, and G. Karypis, "Anomaly detection on attributed networks via contrastive self-supervised learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 6, pp. 2378–2392, Jun. 2022.

[8] J. Wang, Z. Ma, F. Nie, and X. Li, "Fast self-supervised clustering with anchor graph," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 9, pp. 4199–4212, Sep. 2022.

[9] N. T. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. ICLR*, 2017, pp. 1–14.

[10] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. ICML*, 2020, pp. 1597–1607.

[11] J. D. Lee, Q. Lei, N. Saunshi, and J. Zhuo, "Predicting what you already know helps: Provable self-supervised learning," in *Proc. NeurIPS*, 2021, pp. 1–15.

[12] P. Velickovic, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, "Deep graph infomax," in *Proc. ICLR*, 2019, pp. 1–46.

[13] Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, and L. Wang, "Deep graph contrastive representation learning," 2020, *arXiv:2006.04131*.

[14] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," in *Proc. ECCV*, 2020, pp. 776–794.

[15] K. Hassani and A. H. Khasahmadi, "Contrastive multi-view representation learning on graphs," in *Proc. ICML*, 2020, pp. 4116–4126.

[16] T. Gao, X. Yao, and D. Chen, "SimCSE: Simple contrastive learning of sentence embeddings," in *Proc. EMNLP*, 2021, pp. 6894–6910.

[17] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. Hinton, "Big self-supervised models are strong semi-supervised learners," in *Proc. NeurIPS*, 2020, pp. 1–13.

[18] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, "Graph contrastive learning with augmentations," in *Proc. NeurIPS*, 2020, pp. 1–12.

[19] L. Peng, N. Wang, J. Xu, X. Zhu, and X. Li, "GATE: Graph CCA for temporal SElf-supervised learning for label-efficient fMRI analysis," *IEEE Trans. Med. Imag.*, pp. 1–12, 2022.

[20] C. Tang et al., "Feature selective projection with low-rank embedding and dual Laplacian regularization," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 9, pp. 1747–1760, Sep. 2020.

[21] C. Tang et al., "Cross-view locality preserved diversity and consensus learning for multi-view unsupervised feature selection," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 10, pp. 4705–4716, Oct. 2022.

[22] W. Jin et al., "Self-supervised learning on graphs: Deep insights and new direction," 2020, *arXiv:2006.10141*.

[23] X. Liu et al., "Self-supervised learning: Generative or contrastive," 2020, *arXiv:2006.08218*.

[24] J. Zeng and P. Xie, "Contrastive self-supervised learning for graph classification," in *Proc. AAAI*, 2021, pp. 10824–10832.

[25] C. Yuan, Z. Zhong, C. Lei, X. Zhu, and R. Hu, "Adaptive reverse graph learning for robust subspace learning," *Inf. Process. Manage.*, vol. 58, no. 6, Nov. 2021, Art. no. 102733.

[26] Y. Zhu, J. Ma, C. Yuan, and X. Zhu, "Interpretable learning based dynamic graph convolutional networks for Alzheimer's disease analysis," *Inf. Fusion*, vol. 77, pp. 53–61, Jan. 2022.

[27] X. Xu, T. Wang, Y. Yang, A. Hanjalic, and H. T. Shen, "Radial graph convolutional network for visual question generation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 4, pp. 1654–1667, Apr. 2021.

[28] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9729–9738.

[29] J. Devlin, M.-W. Chang, K. Lee, and K. N. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL*, 2018, pp. 1–16.

[30] H. Wang et al., "Self-supervised learning for contextualized extractive summarization," in *Proc. ACL*, Jul. 2019, pp. 2221–2227.

[31] N. Saunshi, O. Plevrakis, S. Arora, M. Khodak, and H. Khandeparkar, "A theoretical analysis of contrastive unsupervised representation learning," in *Proc. ICML*, 2019, pp. 5628–5637.

[32] T. Wang and P. Isola, "Understanding contrastive representation learning through alignment and uniformity on the hypersphere," in *Proc. ICML*, 2020, pp. 9929–9939.

[33] Y.-H. H. Tsai, Y. Wu, R. Salakhutdinov, and L.-P. Morency, "Demystifying self-supervised learning: An information-theoretical framework," 2020, *arXiv:2006.05576*.

[34] C. Tosh, A. Krishnamurthy, and D. Hsu, "Contrastive learning, multi-view redundancy, and linear models," in *Proc. ALT*, 2021, pp. 1179–1206.

[35] F.-Y. Sun, J. Hoffman, V. Verma, and J. Tang, "Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization," in *Proc. ICLR*, 2019, pp. 1–16.

[36] C. Mavromatis and G. Karypis, "Graph InfoClust: Leveraging cluster-level node information for unsupervised graph representation learning," 2020, *arXiv:2009.06946*.

[37] J. Qiu et al., "GCC: Graph contrastive coding for graph neural network pre-training," in *Proc. KDD*, 2020, pp. 1150–1160.

[38] K. Ma et al., "Improving graph representation learning by contrastive regularization," 2021, *arXiv:2101.11525*.

[39] D. Hwang, J. Park, S. Kwon, K. Kim, J.-W. Ha, and H. J. Kim, "Self-supervised auxiliary learning with meta-paths for heterogeneous graphs," in *Proc. NeurIPS*, vol. 33, 2020, pp. 10294–10305.

[40] F. L. Opolka, A. Solomon, C. Cangea, P. Veličković, P. Liò, and R. Devon Hjelm, "Spatio-temporal deep graph infomax," 2019, *arXiv:1904.06316*.

[41] Z. Peng et al., "Graph representation learning via graphical mutual information maximization," in *Proc. WWW*, 2020, pp. 259–270.

[42] J. Zhou et al., "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, Jan. 2020.

[43] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. ICLR*, 2018, pp. 1–12.

[44] Z. Hu, Y. Dong, K. Wang, K. Chang, and Y. Sun, "GPT-GNN: Generative pre-training of graph neural networks," in *Proc. KDD*, 2020, pp. 1857–1867.

[45] A. Subramonian, "Motif-driven contrastive learning of graph representations," in *Proc. AAAI*, 2021, pp. 15980–15981.

[46] P. Khosla et al., "Supervised contrastive learning," in *Proc. NeurIPS*, 2020, pp. 1–13.

[47] A. Iscen, G. Tolias, Y. Avrithis, and O. Chum, "Label propagation for deep semi-supervised learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5070–5079.

[48] H. Wang et al., "Knowledge-aware graph neural networks with label smoothness regularization for recommender systems," in *Proc. KDD*, 2019, pp. 968–977.

[49] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 815–823.

[50] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proc. KDD*, 2014, pp. 701–710.

[51] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, "Collective classification in network data," *AI Mag.*, vol. 29, no. 3, p. 93, 2008.

[52] L. Peng et al., "Reverse graph learning for graph neural network," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Apr. 15, 2022, doi: 10.1109/TNNLS.2022.3161030.

[53] A. Bojchevski and S. Günnemann, "Deep Gaussian embedding of graphs: Unsupervised inductive learning via ranking," in *Proc. ICLR*, 2018, pp. 1–13.

[54] P. Mernyei and C. Cangea, "Wiki-CS: A wikipedia-based benchmark for graph neural networks," 2020, *arXiv:2007.02901*.

[55] B. Rozemberczki, C. Allen, and R. Sarkar, "Multi-scale attributed node embedding," 2019, *arXiv:1909.13021*.

[56] J. McAuley, C. Targett, Q. Shi, and A. van den Hengel, "Image-based recommendations on styles and substitutes," in *Proc. 38th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Aug. 2015, pp. 43–52.

[57] H. Weihua et al., "Open graph benchmark: Datasets for machine learning on graphs," in *Proc. NeurIPS*, 2020, pp. 1–16.

[58] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *ICML*, 2013, vol. 30, no. 1, p. 3.

[59] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. AISTATS*, 2010, pp. 249–256.

[60] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, 2019.

[61] M. Chen, Z. Wei, Z. Huang, B. Ding, and Y. Li, "Simple and deep graph convolutional networks," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1725–1735.

[62] J. Du, S. Zhang, G. Wu, J. M. F. Moura, and S. Kar, "Topology adaptive graph convolutional networks," 2017, *arXiv:1710.10370*.

[63] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. NIPS*, 2017, pp. 1025–1035.

[64] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 1–9.

Liang Peng received the B.S. degree in software from the University of Electronic Science and Technology of China, Chengdu, China, in 2018, and the master's degree from the Intelligent Information Technologies and Applications Laboratory, University of Electronic Science and Technology of China, in 2019, where he is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering.

He was worked with NVIDIA, Shanghai, China, in 2017. His current research interests include graph representation learning and medical image analysis.

Yujie Mo received the B.S. degree in computer science and technology from Northeastern University, Shenyang, China, in 2020. He is currently pursuing the M.S. degree with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China.

His research interests include graph representation learning and related applications.

Jie Xu received the B.Eng. degree from the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China, in 2020, where he is currently pursuing the Ph.D. degree.

His research interests include deep learning, multi-view clustering, and incomplete multi-view clustering.

Jialie Shen (Senior Member, IEEE) received the Ph.D. degree in computer science from the University of New South Wales (UNSW), Sydney NSW, Australia, in 2012, in the area of intelligent media search with large-scale neural networks.

He is currently a Professor in computer vision and machine learning (Chair) with the Department of Computer Vision, City, University of London, London, U.K. His research interests spread across subareas in artificial intelligence (AI), including computer vision, deep learning, image/video analytics, and machine learning.

Dr. Shen's research results have expounded in more than 150 publications at prestigious journals and conferences, with several awards: the Lee Foundation Fellowship for Research Excellence Singapore, the Microsoft Mobile Plus Cloud Computing Theme Research Program Award, the Best Paper Runner-Up for IEEE TRANSACTIONS ON MULTIMEDIA, the Best Reviewer Award for *Information Processing and Management* (IP&M) in 2019 and ACM Multimedia in 2020, and the Test of Time Reviewer Award for *IP&M* in 2022. He serves as an Associate Editor and a member for the Editorial Board for leading journals: *IP&M*, *Pattern Recognition* (PR), IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON MULTIMEDIA, and *ACM Transactions on Multimedia Computing, Communications, and Applications* (ACM TOMM).

Xiaoshuang Shi received the B.S. degree in automation from Northwestern Polytechnical University, Xi'an, China, in 2009, the M.S. degree in automation from Tsinghua University, Beijing, China, in 2013, and the Ph.D. degree from the J. Crayton Pruitt Family Department of Biomedical Engineering, University of Florida, Gainesville, FL, USA, in 2019.

From September 2013 to April 2015, he was a Research Assistant with the Shenzhen Key Laboratory of Broadband Network and Multimedia, Graduate School at Shenzhen, Tsinghua University. His current research interests include large-scale image retrieval, deep learning, and medical image analysis.

Xiaoxiao Li (Member, IEEE) received the Ph.D. degree from Yale University, New Haven, CT, USA, in 2020.

She was a Post-Doctoral Research Fellow with the Computer Science Department, Princeton University, Princeton, NJ, USA. Since August 2021, she has been an Assistant Professor at the Department of Electrical and Computer Engineering (ECE), University of British Columbia (UBC), Vancouver, BC, Canada. In the last few years, she has over 30 papers published in leading machine learning conferences and journals, including NeurIPS, ICML, ICLR, MICCAL, IPML, BMVC, IEEE TRANSACTIONS ON MEDICAL LMAGING, and *Medical lmage Analysis*. Her research interests include range across the interdisciplinary fields of deep learning and biomedical data analysis, aiming to improve the trustworthiness of artificial intelligence (AI) systems for health care.

Dr. Li's work has been recognized with several best paper awards at international conferences.

Heng Tao Shen (Fellow, IEEE) received the B.Sc. (Hons.) and Ph.D. degrees from the Department of Computer Science, National University of Singapore, Singapore, in 2000 and 2004, respectively.

He then joined at the University of Queensland and became a Professor in late 2011. He is a Distinguished Professor and the Dean of School of Computer Science and Engineering, the Executive Dean of artificial intelligence (AI) Research Institute, and the Director of Center for Future Media at the University of Electronic Science and Technology of China, Chengdu. China. He has published more than 350 peer-reviewed articles, including 130 IEEE/ACM TRANSACTIONS. His research interests mainly include multimedia search, computer vision, AI, and big data management.

Dr. Shen is an OSA Fellow and an ACM Fellow. He received seven best paper awards for international conferences, including the Best Paper Award from ACM Multimedia in 2017 and the Best Paper Award Honorable Mention from ACM SIGIR in 2017. He is/was an Associate Editor of *ACM Transactions of Data Science*, IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON MULTIMEDIA, and IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING.

Xiaofeng Zhu (Senior Member, IEEE) received the Ph.D. degree in computer science from The University of Queensland, Brisbane, QLD, Australia, in 2014. He is currently with the University of Electronic Science and Technology of China, Chengdu, China.

His current research interests include big data, artificial intelligence (AI), and medical image analysis.