Deep Graph Contrastive Representation Learning

Yanqiao Zhu[1,2]* **Yichen Xu**[3]* **Feng Yu**[1,2] **Qiang Liu**[4,5] **Shu Wu**[1,2] **Liang Wang**[1,2]
[1] Center for Research on Intelligent Perception and Computing
Institute of Automation, Chinese Academy of Sciences
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences
[3] School of Computer Science, Beijing University of Posts and Telecommunications
[4] RealAI [5] Tsinghua University
yanqiao.zhu@cripac.ia.ac.cn, linyxus@bupt.edu.cn
{feng.yu, shu.wu, wangliang}@nlpr.ia.ac.cn, qiang.liu@realai.ai

Abstract

Graph representation learning nowadays becomes fundamental in analyzing graph-structured data. Inspired by recent success of contrastive methods, in this paper, we propose a novel framework for unsupervised graph representation learning by leveraging a contrastive objective at the node level. Specifically, we generate two graph views by corruption and learn node representations by maximizing the agreement of node representations in these two views. To provide diverse node contexts for the contrastive objective, we propose a hybrid scheme for generating graph views on both structure and attribute levels. Besides, we provide theoretical justification behind our motivation from two perspectives, mutual information and the classical triplet loss. We perform empirical experiments on both transductive and inductive learning tasks using a variety of real-world datasets. Experimental experiments demonstrate that despite its simplicity, our proposed method consistently outperforms existing state-of-the-art methods by large margins. Notably, our method gains about 10% absolute improvements on protein function prediction. Our unsupervised method even surpasses its supervised counterparts on transductive tasks, demonstrating its great potential in real-world applications.

1 Introduction

Over the past few years, graph representation learning has emerged as a powerful strategy for analyzing graph-structured data. Graph representation learning aims to learn an encoding function that transforms nodes to low-dimensional dense embeddings that preserve graph attributive and structural features. Traditional unsupervised graph representation learning approaches, such as DeepWalk [1] and node2vec [2], follow a *contrastive* framework originated in the skip-gram model [3]. Specifically, they first sample short random walks and then enforce neighboring nodes on the same walk to share similar embeddings by contrasting them with other nodes. However, DeepWalk-based methods can be seen as reconstructing the graph proximity matrix, such as high-order adjacent matrix [4], which excessively emphasize proximity information defined on the network structure [5].

Recently, graph representation learning using Graph Neural Networks (GNN) has received considerable attention. Along with its prosperous development, however, there is an increasing concern over the label availability when training the model. Nevertheless, existing GNN models are mostly established in a supervised manner [6–8], which require abundant labeled nodes for training. Albeit with some attempts connecting previous unsupervised objectives (i.e., matrix reconstruction) to GNN models [9, 10], these methods still heavily rely on the preset graph proximity matrix.

*The first two authors contributed equally to this work.

Preprint. Under review.

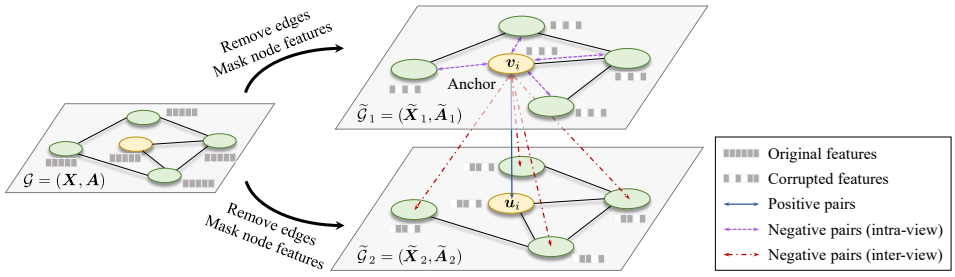


Figure 1: Our proposed deep GRAph Contrastive rEpresentation learning (GRACE) model.

Instead of optimizing the reconstruction objective, visual representation learning leads to revitalization of the classical information maximization (InfoMax) principle [11]. A series of contrastive learning methods have been proposed so far [12–17], which seek to maximize the Mutual Information (MI) between the input (i.e., images) and its representations (i.e., image embeddings) by contrasting positive pairs with negative-sampled counterparts. Inspired by previous success of the Deep InfoMax (DIM) method [15] in visual representation learning, Deep Graph InfoMax (DGI) [18] proposes an alternative objective based on MI maximization in the graph domain. DGI firstly employs GNN to learn node embeddings and obtains a global summary embedding (i.e., the graph embedding), via a readout function. The objective used in DGI is then to maximize the MI between node embeddings and the graph embedding by discriminating nodes in the original graph from nodes in a corrupted graph.

However, we argue that the local-global MI maximization framework in DGI is still in its infancy. Its objective is proved to be equivalent to maximizing the MI between input node features and high-level node embeddings under some conditions. Specifically, to implement the InfoMax objective, DGI requires an injective readout function to produce the global graph embedding, where the injective property is too restrictive to fulfill. For the mean-pooling readout function employed in DGI, it is not guaranteed that the graph embedding can distill useful information from nodes, as it is insufficient to preserve distinctive features from node-level embeddings. Moreover, DGI proposes to use feature shuffling to generate corrupted views of graphs. Nevertheless, this scheme considers corrupting node features at a coarse-grained level when generating negative node samples. When the feature matrix is sparse, performing feature shuffling only is insufficient to generate different neighborhoods (i.e., contexts) for nodes in the corrupted graph, leading to difficulty in learning of the contrastive objective.

In this paper, we introduce a simple yet powerful contrastive framework for unsupervised graph representation learning (Figure 1), which we refer to as deep GRAph Contrastive rEpresentation learning (GRACE), motivated by a traditional self-organizing network [19] and its recent renaissance in visual representation learning [17]. Rather than contrasting node-level embeddings to global ones, we primarily focus on contrasting embeddings at the node level and our work makes no assumptions on injective readout functions for generating the graph embedding. In GRACE, we first generate two correlated *graph views* by randomly performing *corruption*. Then, we train the model using a contrastive loss to maximize the agreement between node embeddings in these two views. Unlike visual data, where abundant image transformation techniques are available, how to perform corruption to generate views for graphs is still an open problem. In our work, we jointly consider corruption at both topology and node attribute levels, namely removing edges and masking features, to provide diverse contexts for nodes in different views, so as to boost optimization of the contrastive objective. Last, we provide theoretical analysis that reveals the connections from our contrastive objective to mutual information and the classical triplet loss.

Our contribution is summarized as follows. Firstly, we propose a general contrastive framework for unsupervised graph representation learning. The proposed GRACE framework simplifies previous work and works by maximizing the agreement of node embeddings between two graph views. Secondly, we propose two specific schemes, removing edges and masking features, to generate views of graphs. Finally, we conduct comprehensive empirical studies using six popular public benchmark datasets on both transductive and inductive node classification under the commonly-used linear evaluation protocol. GRACE consistently outperforms existing methods and achieves unsupervised

state-of-the-art performance on protein function prediction (about 10% absolute improvement). In addition, our unsupervised method even surpasses its supervised counterparts on transductive tasks, demonstrating its great potential in real-world applications.

2 Related Work

Contrastive learning of visual representations. Being popular in self-supervised visual representation learning, contrastive methods aim to learn discriminative representations by contrasting positive and negative samples. For visual data, negative samples can be generated using image augmentation techniques such as cropping, rotation [20], color distortion [21], etc. Existing work [12–14] employs a memory bank for storing negative samples. Other work [15–17] explores in-batch negative samples. For an image patch as the anchor, these methods usually find a global summary vector [22, 15] or patches in neighboring views [23, 24] as the positive sample, and contrast them with negative-sampled counterparts, such as patches of other images within the same batch [22].

Theoretical analysis sheds light on the reasons behind their success [25]. Objectives used in these methods can be seen as maximizing the lower bounds of MI between input features and their representations [11]. However, recent work [26] reveals that downstream performance in evaluating the quality of representations may strongly depend on the bias that is encoded not only in the convolutional architectures but also in the specific estimator of the InfoMax objective.

Graph representation learning. Many traditional methods on unsupervised graph representation learning employ the contrastive paradigm as well [1, 2, 9, 27]. Prior work on unsupervised graph representation learning focuses on local contrastive patterns, which forces neighboring nodes to have similar embeddings. Positive samples under this circumstance are nodes appearing in the same random walk [1, 2]. For example, the pioneering work DeepWalk [1] models probabilities of node co-occurrence pairs using noise-contrastive estimation [28]. These random-walk-based methods are proved to be equivalent to factorizing some forms of graph proximity (e.g., transformation of the adjacent matrix) [4], which overly emphasize on the structural information encoded in these graph proximities and also face severe scaling problem with large-scale datasets. Also, these methods are known to be error-prone with inappropriate hyperparameter tuning [1, 2].

Recent work on graph neural networks (GNN) employs more powerful graph convolutional encoders over conventional methods. Among them, considerable literature has grown up around the theme of supervised GNN [6–8, 29], which requires labeled datasets that may not be accessible in real-world applications. Along the other line of development, unsupervised GNNs receive little attention. Representative methods include GraphSAGE [10], which incorporates DeepWalk-like objectives as well. Recent work DGI [18] marries the power of GNN and contrastive learning, which focuses on maximizing MI between global graph embeddings and local node embeddings. However, it is hard to fulfill the injective requirement of the graph readout function such that the graph embedding may be deteriorated. In contrast to DGI, our work does not rely on an explicit graph embedding. Instead, we focus on maximizing the agreement of node embeddings across two corrupted views of the graph.

3 Deep Graph Contrastive Representation Learning

In this section, we present our proposed GRACE framework in detail, starting with the overall framework of contrastive objectives, followed by specific graph view generation methods. At the end of this section, we provide theoretical justification behind our framework from two perspectives, i.e., connection to the InfoMax principle and the classical triplet loss.

3.1 Preliminaries

In unsupervised graph representation learning, let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote a graph, where $\mathcal{V} = \{v_1, v_2, \cdots, v_N\}$, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ represent the node set and the edge set respectively. We denote the feature matrix and the adjacency matrix as $\boldsymbol{X} \in \mathbb{R}^{N \times F}$ and $\boldsymbol{A} \in \{0, 1\}^{N \times N}$, where $\boldsymbol{x}_i \in \mathbb{R}^F$ is the feature of v_i, and $\boldsymbol{A}_{ij} = 1$ iff $(v_i, v_j) \in \mathcal{E}$. There is no given class information of nodes in $\mathcal{G}$ during training. Our objective is to learn a GNN encoder $f(\boldsymbol{X}, \boldsymbol{A}) \in \mathbb{R}^{N \times F'}$ receiving the graph features and structure as input, that produces node embeddings in low dimensionality, i.e., $F' \ll F$.

We denote $\boldsymbol{H} = f(\boldsymbol{X}, \boldsymbol{A})$ as the learned representations of nodes, where $\boldsymbol{h}_i$ is the embedding of node v_i. These representations can be used in downstream tasks, such as node classification.

3.2 Contrastive Learning of Node Representations

3.2.1 The Contrastive Learning Framework

Contrary to previous work that learns representations by utilizing local-global relationships, in GRACE, we learn embeddings by directly maximizing node-level agreement between embeddings. To be specific, we first generate two graph views by randomly corrupting the original graph. Then, we employ a contrastive objective that enforces the encoded embeddings of each node in the two different views agree with each other and can be distinguished from embeddings of other nodes.

In our GRACE model, at each iteration, we generate two graph views, denoted as $\widetilde{G}_1$ and $\widetilde{G}_2$, and denote node embeddings in the two generated views as $\boldsymbol{U} = f(\widetilde{\boldsymbol{X}}_1, \widetilde{\boldsymbol{A}}_1)$ and $\boldsymbol{V} = f(\widetilde{\boldsymbol{X}}_2, \widetilde{\boldsymbol{A}}_2)$, where $\widetilde{\boldsymbol{X}}_*$ and $\widetilde{\boldsymbol{A}}_*$ are the feature matrices and adjacent matrices of the views. Details on the generation of graph views will be discussed later in Section 3.2.2.

Then, we employ a contrastive objective (i.e., a discriminator) that distinguishes the embeddings of the same node in these two different views from other node embeddings. For any node v_i, its embedding generated in one view, $\boldsymbol{u}_i$, is treated as the anchor, the embedding of it generated in the other view, $\boldsymbol{v}_i$, forms the positive sample, and embeddings of nodes other than v_i in the two views are naturally regarded as negative samples. Formally, we define the critic $\theta(\boldsymbol{u}, \boldsymbol{v}) = s(g(\boldsymbol{u}), g(\boldsymbol{v}))$, where s is the cosine similarity and g is a non-linear projection to enhance the expression power of the critic [17, 26]. The projection g is implemented with a two-layer multilayer perceptron (MLP). We define the pairwise objective for each positive pair $(\boldsymbol{u}_i, \boldsymbol{v}_i)$ as

$$\ell(\boldsymbol{u}_i, \boldsymbol{v}_i) = \log \frac{e^{\theta(\boldsymbol{u}_i, \boldsymbol{v}_i)/\tau}}{\underbrace{e^{\theta(\boldsymbol{u}_i, \boldsymbol{v}_i)/\tau}}_{\text{the positive pair}} + \underbrace{\sum_{k=1}^{N} \mathbb{1}_{[k \neq i]} e^{\theta(\boldsymbol{u}_i, \boldsymbol{v}_k)/\tau}}_{\text{inter-view negative pairs}} + \underbrace{\sum_{k=1}^{N} \mathbb{1}_{[k \neq i]} e^{\theta(\boldsymbol{u}_i, \boldsymbol{u}_k)/\tau}}_{\text{intra-view negative pairs}}}, \tag{1}$$

where $\mathbb{1}_{[k \neq i]} \in \{0, 1\}$ is an indication function that equals to 1 iff $k \neq i$, and τ is a temperature parameter. Please note that, in our work, we do not sample negative nodes *explicitly*. Instead, given a positive pair, we naturally define negative samples as all other nodes in the two views. Therefore, negative samples come from two sources, inter-view or intra-view nodes, corresponding to the second and the third term in the denominator, respectively. Since two views are symmetric, the loss for another view is defined similarly for $\ell(\boldsymbol{v}_i, \boldsymbol{u}_i)$. The overall objective to be maximized is then defined as the average over all positive pairs, formally given by

$$\mathcal{J} = \frac{1}{2N} \sum_{i=1}^{N} \left[\ell(\boldsymbol{u}_i, \boldsymbol{v}_i) + \ell(\boldsymbol{v}_i, \boldsymbol{u}_i)\right]. \tag{2}$$

To sum up, at each training epoch, GRACE first generates two graph views $\widetilde{\mathcal{G}}_1$ and $\widetilde{\mathcal{G}}_2$ of graph $\mathcal{G}$. Then, we obtain node representations $\boldsymbol{U}$ and $\boldsymbol{V}$ of $\widetilde{\mathcal{G}}_1$ and $\widetilde{\mathcal{G}}_2$ using a GNN encoder f. Finally, the parameters of f and g is updated by maximizing the objective in Eq. (2). The learning algorithm is summarized in Algorithm 1.

Algorithm 1: GRACE training algorithm

for $epoch \leftarrow 1, 2, \cdots$ **do**
- Generate two graph views $\widetilde{\mathcal{G}}_1$ and $\widetilde{\mathcal{G}}_2$ by performing corruption on $\mathcal{G}$
- Obtain node embeddings $\boldsymbol{U}$ of $\widetilde{\mathcal{G}}_1$ using the encoder f
- Obtain node embeddings $\boldsymbol{V}$ of $\widetilde{\mathcal{G}}_2$ using the encoder f
- Compute the contrastive objective $\mathcal{J}$ with Eq. (2)
- Update parameters by applying stochastic gradient ascent to maximize $\mathcal{J}$

3.2.2 Graph View Generation

Generating views is a key component of contrastive learning methods. In the graph domain, different views of a graph provide different contexts for each node. Considering contrastive approaches that rely on contrasting between node embeddings in different views, we propose to corrupt the original graph at both structure and attribute levels, which constructs diverse node contexts for the model to contrast with. In GRACE, we design two methods for graph corruption, removing edges for topology and masking features for node attributes. How to perform graph corruption is still an open problem [18]. It is flexible to adopt other alternative mechanisms of corruption methods in our framework.

Removing edges (RE). We randomly remove a portion of edges in the original graph. Formally, since we only remove existing edges, we first sample a random masking matrix $\widetilde{\boldsymbol{R}} \in \{0,1\}^{N\times N}$, whose entry is drawn from a Bernoulli distribution $\widetilde{\boldsymbol{R}}_{ij} \sim \mathcal{B}(1-p_r)$ if $\boldsymbol{A}_{ij}=1$ for the original graph and $\widetilde{\boldsymbol{R}}_{ij}=0$ otherwise. Here p_r is the probability of each edge being removed. The resulting adjacency matrix can be computed as

$$\widetilde{\boldsymbol{A}} = \boldsymbol{A} \circ \widetilde{\boldsymbol{R}}, \tag{3}$$

where $(\boldsymbol{x}\circ\boldsymbol{y})_i = x_i y_i$ is Hadamard product.

Masking node features (MF). Apart from removing edges, we randomly mask a fraction of dimensions with zeros in node features. Formally, we first sample a random vector $\widetilde{\boldsymbol{m}} \in \{0,1\}^F$ where each dimension of it independently is drawn from a Bernoulli distribution with probability $1-p_m$, i.e., $\widetilde{m}_i \sim \mathcal{B}(1-p_m), \forall i$. Then, the generated node features $\widetilde{\boldsymbol{X}}$ is computed by

$$\widetilde{\boldsymbol{X}} = [\boldsymbol{x}_1 \circ \widetilde{\boldsymbol{m}}; \boldsymbol{x}_2 \circ \widetilde{\boldsymbol{m}}; \cdots; \boldsymbol{x}_N \circ \widetilde{\boldsymbol{m}}]^\top. \tag{4}$$

Here $[\cdot;\cdot]$ is the concatenation operator.

Please kindly note that although our proposed RE and MF schemes are technically similar to Dropout [30] and DropEdge [31], our GRACE model and the two referred methods are proposed for fundamentally different purposes. Dropout is a general technique that randomly masks neurons during training to prevent over-fitting of large-scale models. In the graph domain, DropEdge is proposed to prevent over-fitting and alleviate over-smoothing *when the GNN architecture is too deep*. However, our GRACE framework randomly applies RE and MF to produce different graph views for contrastive learning at both graph topology and node feature levels. Moreover, the employed GNN encoder in GRACE is a rather shallow model, usually consisting of only two or three layers.

In our implementation, we jointly leverage these two methods to generate graph views. The generation of $\widetilde{\mathcal{G}}_1$ and $\widetilde{\mathcal{G}}_2$ are controlled by two hyperparameters p_r and p_m. To provide different contexts in the two views, the generation process of the two views uses two different sets of hyperparameters $p_{r,1}, p_{m,1}$ and $p_{r,2}, p_{m,2}$. Experiments demonstrate that our model is not sensitive to the choice of p_r and p_m under mild conditions such that the original graph is not overly corrupted, e.g., $p_r \leq 0.8$ and $p_m \leq 0.8$. We refer readers to the sensitivity analysis presented in Appendix C.1 for empirical results.

3.3 Theoretical Justification

In this section, we provide theoretical justification behind our model from two perspectives, i.e., the mutual information maximization and the triplet loss. Detailed proofs can be found in Appendix D.

Connections to the mutual information. Firstly, we reveal the connection between our loss and mutual information maximization between node features and the embeddings in the two views, which has been widely applied in the representation learning literature [13, 15, 25, 26]. MI quantifies the amount of information obtained about one random variable by observing the other random variable.

Theorem 1. *Let $\mathbf{X}_i = \{\boldsymbol{x}_k\}_{k\in\mathcal{N}(i)}$ be the neighborhood of node v_i that collectively maps to its output embedding, where $\mathcal{N}(i)$ denotes the set of neighbors of node v_i specified by GNN architectures, and $\mathbf{X}$ be the corresponding random variable with a uniform distribution $p(\mathbf{X}_i)=\frac{1}{N}$. Given two random variables $\mathbf{U},\mathbf{V}\in\mathbb{R}^{F'}$ being the embedding in the two views, with their joint distribution denoted as $p(\mathbf{U},\mathbf{V})$, our objective $\mathcal{J}$ is a lower bound of MI between encoder input $\mathbf{X}$ and node representations in two graph views $\mathbf{U},\mathbf{V}$. Formally,*

$$\mathcal{J} \leq I(\mathbf{X};\mathbf{U},\mathbf{V}). \tag{5}$$

Proof sketch. We first observe that our objective $\mathcal{J}$ is a lower bound of the InfoNCE objective [23], which is defined as $I_{\text{NCE}}(\mathbf{U};\mathbf{V}) \triangleq \mathbb{E}_{\prod_i p(\boldsymbol{u}_i,\boldsymbol{v}_i)}\left[\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{\theta(\boldsymbol{u}_i,\boldsymbol{v}_i)}}{\frac{1}{N}\sum_{j=1}^{N}e^{\theta(\boldsymbol{u}_i,\boldsymbol{v}_j)}}\right]$ [25]. According to [23], the InfoNCE estimator is a lower bound of the true MI. Therefore, the theorem directly follows from the application of data processing inequality, which states that $I(\mathbf{U};\mathbf{V}) \leq I(\mathbf{X};\mathbf{U},\mathbf{V})$. □

Remark. From Theorem 1, it reveals that maximizing $\mathcal{J}$ is equivalent to maximizing a lower bound of the mutual information $I(\mathbf{X};\mathbf{U},\mathbf{V})$ between input node features and learned node representations. Counterintuitively, recent work further provides empirical evidence that optimizing a stricter bound of MI may not lead to better downstream performance on visual representation learning [26], which highlights the importance of the encoder design. In Appendix C.3, we also compare our objective with the InfoNCE loss, which is a stricter estimator of MI, to further demonstrate the superiority of the GRACE model.

Connections to the triplet loss. Alternatively, we may view the optimization problem in Eq. (2) as a classical triplet loss, commonly used in deep metric learning.

Theorem 2. *When the projection function g is the identity function and we measure embedding similarity by simply taking inner product, i.e., $s(\boldsymbol{u},\boldsymbol{v}) = \boldsymbol{u}^\top\boldsymbol{v}$, and further assuming that positive pairs are far more aligned than negative pairs, minimizing the pairwise objective $\ell(\boldsymbol{u}_i,\boldsymbol{v}_i)$ coincides with maximizing the triplet loss, as given in the sequel*

$$-\ell(\boldsymbol{u}_i,\boldsymbol{v}_i) \propto 4N\tau + \sum_{j=1}^{N}\mathbb{1}_{[j\neq i]}\left[\left(\|\boldsymbol{u}_i-\boldsymbol{v}_i\|^2 - \|\boldsymbol{u}_i-\boldsymbol{v}_j\|^2\right) + \left(\|\boldsymbol{u}_i-\boldsymbol{v}_i\|^2 - \|\boldsymbol{u}_i-\boldsymbol{u}_j\|^2\right)\right]. \tag{6}$$

Remark. Theorem 2 draws connection between the objective and the classical triplet loss. In other words, we may regard the problem in Eq. (2) as learning graph convolutional encoders to encourage positive samples being further away from negative samples in the embedding space. Moreover, by viewing the objective from the metric learning perspective, we highlight the importance of appropriately choosing negative samples, which is often neglected in previous InfoMax-based methods. Last, the contrastive objective is cheap to optimize since we do not have to generate negative samples explicitly and all computation can be performed in parallel. In contrast, the triplet loss is known to be computationally expensive [32].

4 Experiments

In this section, we empirically evaluate the quality of produced node embeddings on node classification using six public benchmark datasets. We refer readers of interest to the supplementary material on details of experiments, including dataset configurations (Appendix A), implementation and hyperparameters (Appendix B), and additional experiments (Appendix C).

4.1 Datasets

For comprehensive comparison, we use six widely-used datasets to study the performance of both transductive and inductive node classification. Specifically, we use three kinds of datasets: (1) citation networks including Cora, Citeseer, Pubmed, and DBLP [33, 34] for transductive node classification, (2) social networks from Reddit posts for inductive learning on large-scale graphs [10], and (3) biological protein-protein interaction (PPI) networks [35] for inductive node classification on multiple graphs. Details of these datasets can be found in Appendix A.

4.2 Experimental Setup

For every experiment, we follow the linear evaluation scheme as in [18], where each model is firstly trained in an unsupervised manner. The resulting embeddings are used to train and test a simple ℓ_2-regularized logistic regression classifier. We train the model for twenty runs and report the averaged performance on each dataset. Moreover, we measure performance using micro-averaged F1-score on inductive tasks and accuracy on transductive tasks. Please kindly note that for inductive learning tasks, tests are conducted on unseen or untrained nodes and graphs, while for transductive learning tasks, we use the features of all data, but the labels of the test set are masked during training.

Transductive learning. In transductive learning tasks, we employ a two-layer GCN [6] as the encoder. Our encoder architecture is formally given by

$$\mathrm{GC}_i(\boldsymbol{X}, \boldsymbol{A}) = \sigma\left(\hat{\boldsymbol{D}}^{-\frac{1}{2}}\hat{\boldsymbol{A}}\hat{\boldsymbol{D}}^{-\frac{1}{2}}\boldsymbol{X}\boldsymbol{W}_i\right), \qquad f(\boldsymbol{X}, \boldsymbol{A}) = \mathrm{GC}_2(\mathrm{GC}_1(\boldsymbol{X}, \boldsymbol{A}), \boldsymbol{A}). \tag{7}$$

where $\hat{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}$ is the adjacency matrix with self-loops, $\hat{\boldsymbol{D}} = \sum_i \hat{\boldsymbol{A}}_i$ is the degree matrix, $\sigma(\cdot)$ is a nonlinear activation function, e.g., $\mathrm{ReLU}(\cdot) = \max(0, \cdot)$, and $\boldsymbol{W}$ is a trainable weight matrix.

We consider the following two categories of representative algorithms as baselines, including (1) traditional methods DeepWalk [1] and node2vec [2], and (2) deep learning methods GAE, VGAE [9], and DGI [18]. Furthermore, we report performance obtained using a logistic regression classifier on raw node features and DeepWalk with embeddings concatenated with input node features. For direct comparison with supervised counterparts, we also report the performance of two representative models SGC [29] and GCN [6], where they are trained in an end-to-end fashion.

Inductive learning on large graphs. Considering the large scale of the Reddit data, we closely follow [18] and employ a three-layer GraphSAGE-GCN [10] with residual connections [36] as the encoder, which is formulated as

$$\widehat{\mathrm{MP}}_i(\boldsymbol{X}, \boldsymbol{A}) = \sigma([\hat{\boldsymbol{D}}^{-1}\hat{\boldsymbol{A}}\boldsymbol{X}; \boldsymbol{X}]\boldsymbol{W}_i), \qquad f(\boldsymbol{X}, \boldsymbol{A}) = \widehat{\mathrm{MP}}_3(\widehat{\mathrm{MP}}_2(\widehat{\mathrm{MP}}_1(\boldsymbol{X}, \boldsymbol{A}), \boldsymbol{A}), \boldsymbol{A}). \tag{8}$$

Here we use the mean-pooling propagation rule, as $\hat{\boldsymbol{D}}^{-1}$ averages over node features. Due to the large scale of Reddit, it cannot fit into GPU memory entirely. Therefore, we apply the subsampling method proposed in [10], where we first randomly select a minibatch of nodes, then a subgraph centered around each selected node is obtained by sampling node neighbors with replacement. To be specific, we sample 30, 25, 20 neighbors at the first-, second-, and third-hop respectively. For generating graph views under such sampling-based settings, both RE and MF can be adapted to sampled subgraphs effortlessly.

Inductive learning on multiple graphs. For inductive learning on multiple graphs PPI, we also apply the mean-pooling propagation rule with GraphSAGE-GCN, using the same setting as Reddit. Since the PPI dataset consists of multiple graphs, we only compute negative samples for one anchor node as other nodes within the same graph, due to efficiency considerations.

Baselines in both large graphs and multiple graphs settings are selected similarly to transductive tasks. We consider (1) traditional methods DeepWalk[2] [1], and (2) deep learning methods GraphSAGE [10] and DGI [18]. Additionally, we report the performance of using raw features and DeepWalk + features under the same settings as in transductive tasks. We further provide the performance of two representative supervised methods, including FastGCN [37] and GaAN-mean [38] for reference. In the table, results of baselines are reported in accordance with performance in their original papers. For GraphSAGE, we reuse the unsupervised results for fair comparison.

4.3 Results and Analysis

The empirical performance is summarized in Table 1. Overall, from the table, we can see that our proposed model shows strong performance across all six datasets. GRACE consistently performs better than unsupervised baselines by considerable margins on both transductive and inductive tasks. The strong performance verifies the superiority of the proposed contrastive learning framework. We particularly note that GRACE is competitive with models *trained with label supervision* on all four transductive datasets and inductive dataset Reddit.

We make other observations as follows. Firstly, GRACE achieves over 10% absolute improvement over another competitive contrastive learning method DGI on PPI. We believe that this is due to the extreme sparsity of node features (over 40% nodes having all-zero features [10]), which emphasizes the importance of considering topological information when choosing negative samples. For datasets like PPI, extreme feature sparsity prevents DGI from discriminating samples in the original graph from the corrupted graph, generated via shuffling node features, since shuffling node features makes no effect for the contrastive objective. Contrarily, the RE scheme used in GRACE does not rely on node features and acts as a remedy under such circumstances, which can explain the large gain of GRACE on PPI compared with DGI. Also, we note that there is still a huge gap between our method

[2]DeepWalk is not applicable to the multi-graph experiments, since the embedding spaces produced by DeepWalk may be arbitrarily rotated with respect to different disjoint graphs [10].

Table 1: Summary of performance on node classification in terms of accuracy in percentage (on transductive tasks) or micro-averaged F1 score (on inductive tasks) with standard deviation. Available data for each method during the training phase is shown in the second column, where $\boldsymbol{X}, \boldsymbol{A}, \boldsymbol{Y}$ correspond to node features, the adjacency matrix, and labels respectively. The highest performance of unsupervised models is highlighted in boldface.

(a) *Transductive*

Method	Training Data	Cora	Citeseer	Pubmed	DBLP
Raw features	$\boldsymbol{X}$	64.8	64.6	84.8	71.6
node2vec	$\boldsymbol{A}$	74.8	52.3	80.3	78.8
DeepWalk	$\boldsymbol{A}$	75.7	50.5	80.5	75.9
DeepWalk + features	$\boldsymbol{X}, \boldsymbol{A}$	73.1	47.6	83.7	78.1
GAE	$\boldsymbol{X}, \boldsymbol{A}$	76.9	60.6	82.9	81.2
VGAE	$\boldsymbol{X}, \boldsymbol{A}$	78.9	61.2	83.0	81.7
DGI	$\boldsymbol{X}, \boldsymbol{A}$	82.6±0.4	68.8±0.7	86.0±0.1	83.2±0.1
GRACE	$\boldsymbol{X}, \boldsymbol{A}$	**83.3±0.4**	**72.1±0.5**	**86.7±0.1**	**84.2±0.1**
SGC	$\boldsymbol{X}, \boldsymbol{A}, \boldsymbol{Y}$	80.6	69.1	84.8	81.7
GCN	$\boldsymbol{X}, \boldsymbol{A}, \boldsymbol{Y}$	82.8	72.0	84.9	82.7

(b) *Inductive*

Method	Training Data	Reddit	PPI
Raw features	$\boldsymbol{X}$	58.5	42.2
DeepWalk	$\boldsymbol{A}$	32.4	—
DeepWalk + features	$\boldsymbol{X}, \boldsymbol{A}$	69.1	—
GraphSAGE-GCN	$\boldsymbol{X}, \boldsymbol{A}$	90.8	46.5
GraphSAGE-mean	$\boldsymbol{X}, \boldsymbol{A}$	89.7	48.6
GraphSAGE-LSTM	$\boldsymbol{X}, \boldsymbol{A}$	90.7	48.2
GraphSAGE-pool	$\boldsymbol{X}, \boldsymbol{A}$	89.2	50.2
DGI	$\boldsymbol{X}, \boldsymbol{A}$	94.0±0.1	63.8±0.2
GRACE	$\boldsymbol{X}, \boldsymbol{A}$	**94.2±0.0**	**73.6±0.1**
FastGCN	$\boldsymbol{X}, \boldsymbol{A}, \boldsymbol{Y}$	93.7	—
GaAN-mean	$\boldsymbol{X}, \boldsymbol{A}, \boldsymbol{Y}$	95.8±0.1	96.9±0.2

with supervised models. These supervised models benefit another merit from labels, which provide other auxiliary information for model learning. Considering the sparse nature of real-world datasets, we perform another experiment to verify that our method is robust to sparse node features (Appendix C.4). Results show that by randomly removing node features, our still outperforms existing baselines.

Secondly, the performance of traditional contrastive learning methods like DeepWalk is inferior to the naive classifier that only uses raw features on some datasets (Citeseer, Pubmed, and Reddit), which suggests that these methods may be ineffective in utilizing node features. Unlike traditional work, we see that GCN-based methods, e.g., GraphSAGE and GAE, are capable of incorporating node features when learning embeddings. However, we note that on certain datasets (Pubmed), their performance is still worse than DeepWalk + feature, which we believe can be attributed to their naive method of selecting negative samples that simply chooses contrastive pairs based on edges. This fact further demonstrates the important role of selecting negative samples in contrastive representation learning. The superior performance of GRACE compared to GAEs also once again verifies the effectiveness of our proposed GRACE framework that contrasts nodes across graph views.

Additionally, we perform sensitivity analysis on critical hyperparameters p_r and p_m (Appendix C.1) as well as ablation studies on our hybrid scheme on generating graph views (Appendix C.2). Results show that our method is stable to perturbation of these parameters and verify the necessity of corruption at both graph topology and node feature levels. We also compare the classical InfoNCE loss (Appendix C.3), verifying the efficacy of our design choice. Details of these extra experiments can be found in the supplementary material.

5 Conclusion

In this paper, we have developed a novel graph contrastive representation learning framework based on maximizing the agreement at the node level. Our model learns representations by first generating graph views using two proposed schemes, removing edges and masking node features, and then applying a contrastive loss to maximize the agreement of node embeddings in these two views. Theoretical analysis reveals the connections from our contrastive objective to mutual information maximization and the classical triplet loss, which justifies our motivation. We have conducted comprehensive experiments using various real-world datasets under transductive and inductive settings. Experimental results demonstrate that our proposed method can consistently outperform existing state-of-the-art methods by large margins and even surpass supervised counterparts on transductive tasks.

Discussions on Broader Impact

Our proposed self-supervised graph representation learning techniques help alleviate the label scarcity issue when deploying machine learning applications in real-world, which saves a lot of efforts on human annotating. For example, our GRACE framework can be plugged into existing recommender systems and produces high-quality embeddings for users and commodities to resolve the cold start problem. Moreover, from the empirical results, our work outperforms existing baselines on protein function prediction by significant margins, which demonstrate its great potential in drug discovery and treatment, given the COVID-19 crisis at this critical juncture. Note that our work mainly serves as a plug-in for existing machine learning models, it does not bring new ethical concerns. However, the GRACE model may still give biased outputs (e.g., gender bias, ethnicity bias), as the provided data itself may be strongly biased during the processes of the data collection, graph construction, etc.

Acknowledgements

The authors would like to thank Tao Sun and Sirui Lu for insightful discussions. This work is jointly supported by National Key Research and Development Program (2018YFB1402600, 2016YFB1001000) and National Natural Science Foundation of China (U19B2038, 61772528).

A Dataset Details

Transductive learning. We utilize four widely-used citation networks, Cora, Citeseer, Pubmed, and DBLP, for predicting article subject categories. In these datasets, graphs are constructed from computer science article citation links. Specifically, nodes correspond to articles and undirected edges to citation links between articles. Furthermore, each node has a sparse bag-of-words feature and a corresponding label of article types. The former three networks are provided by [33, 39] and the latter DBLP dataset is provided by [34]. On these citation networks, we randomly select 10% of the nodes as the training set, 10% nodes as the validation set, and leave the rest nodes as the test set.

Inductive learning on large graphs. We then predict community structures of a large-scale social network, collected from Reddit. The dataset, preprocessed by [10], contains Reddit posts created in September 2014, where posts belong to different communities (subreddit). In the dataset, nodes correspond to posts, and edges connect posts if the same user has commented on both. Node features are constructed from post title, content, and comments, using off-the-shelf GloVe word embeddings [40], along with other metrics such as post score and the number of comments. Following the inductive setting of [10, 18], on the Reddit dataset, we choose posts in the first 20 days for training, including 151,708 nodes, and the remaining for testing (with 30% data including 23,699 nodes for validation).

Inductive learning on multiple graphs. Last, we predict protein roles, in terms of their cellular functions from gene ontology, within the protein-protein interaction (PPI) networks [35] to evaluate the generalization ability of the proposed method across multiple graphs. The PPI dataset contains multiple graphs, with each corresponding to a human tissue. The graphs are constructed by [10], where each node has multiple labels that is a subset of gene ontology sets (121 in total), and node features include positional gene sets, motif gene sets, and immunological signatures (50 in total). Following [10], we select twenty graphs consisting of 44,906 nodes as the training set, two graphs containing 6,514 nodes as the validation, and the rest four graphs containing 12,038 nodes as the test set.

The statistics of datasets are summarized in Table 2; download links are included in Table 3. For transductive tasks, similar to [6], during the training phase, all node features are visible but node labels are masked. In the inductive setting, we closely follow [10]; during training, nodes for evaluation are completely invisible; evaluation is then conducted on unseen or untrained nodes and graphs.

Table 2: Statistics of datasets used in experiments.

Dataset	Type	#Nodes	#Edges	#Features	#Classes
Cora	Transductive	2,708	5,429	1,433	7
Citeseer	Transductive	3,327	4,732	3,703	6
Pubmed	Transductive	19,717	44,338	500	3
DBLP	Transductive	17,716	105,734	1,639	4
Reddit	Inductive	231,443	11,606,919	602	41
PPI	Inductive	56,944 (24 graphs)	818,716	50	121 (multilabel)

Table 3: Dataset download links.

Dataset	Download link
Cora	https://github.com/kimiyoung/planetoid/raw/master/data
Citeseer	https://github.com/kimiyoung/planetoid/raw/master/data
Pubmed	https://github.com/kimiyoung/planetoid/raw/master/data
DBLP	https://github.com/abojchevski/graph2gauss/raw/master/data/dblp.npz
Reddit	https://s3.us-east-2.amazonaws.com/dgl.ai/dataset/reddit.zip
PPI	https://s3.us-east-2.amazonaws.com/dgl.ai/dataset/ppi.zip

B Implementation

Computing infrastructures. All models are implemented using PyTorch Geometric 1.5.0 [41] and PyTorch 1.4.0 [42]. All datasets used throughout experiments are available in PyTorch Geometric libraries. For node classification, we use the existing implementation of logistic regression with ℓ_2 regularization from Scikit-learn [43]. All experiments are conducted on a computer server with eight NVIDIA Titan Xp GPUs (12GB memory each) and fourteen Intel Xeon E5-2660 v4 CPUs.

Hyperparameters. All models are initialized with Glorot initialization [44], and trained using Adam SGD optimizer [45] on all datasets. The initial learning rate is set to 0.001 with an exception to 0.0005 on Cora and 10^{-5} on Reddit. The ℓ_2 weight decay factor is set to 10^{-5} on all datasets. On both transductive and inductive tasks, we train the model for a fixed number of epochs, specifically 200, 200, 1500, 1000 epochs for Cora, Citeseer, Pubmed and DBLP, respectively, 40 for Reddit and 200 for PPI. The probability parameters controlling the sampling process, $p_{r,1}, p_{m,1}$ for the first view and $p_{r,2}, p_{m,2}$ for the second view, are all selected between 0.0 and 0.4, since the original graph will be overly corrupted when the probability is set too large. Note that to generate different contexts for nodes in the two views, $p_{r,1}$ and $p_{r,2}$ should be distinct, and the same holds for $p_{m,1}$ and $p_{m,2}$. All dataset-specific hyperparameters are summarized in Table 4.

Table 4: Hypeparameter specifications.

Dataset	$p_{m,1}$	$p_{m,2}$	$p_{r,1}$	$p_{r,2}$	Learning rate	Weight decay	Training epochs	Hidden dimension	Activation function
Cora	0.3	0.4	0.2	0.4	0.005	10^{-5}	200	128	ReLU
Citeseer	0.3	0.2	0.2	0.0	0.001	10^{-5}	200	256	PReLU
Pubmed	0.0	0.2	0.4	0.1	0.001	10^{-5}	1,500	256	ReLU
DBLP	0.1	0.0	0.1	0.4	0.001	10^{-5}	1,000	256	ReLU
Reddit	0.3	0.2	0.1	0.2	0.00001	10^{-5}	40	512	ELU
PPI	0.3	0.4	0.2	0.3	0.001	10^{-5}	200	128	ReLU

C Additional Experiments

C.1 Sensitivity Analysis

In this section, we perform sensitivity analysis on critical hyperparameters in GRACE, namely four probabilities $p_{m,1}, p_{r,1}, p_{m,2}, p_{r,2}$ that determine the generation of graph views to show the model stability under the perturbation of these hyperparameters. We conduct trasductive node classification by varying these parameters from 0.1 to 0.9. For sake of visualization brevity, we set $p_1 = p_{r,1} = p_{m,1}$

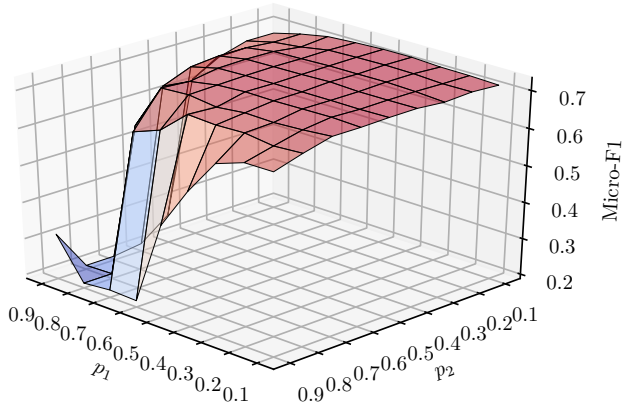


Figure 2: The performance of GRACE with varying different hyperparameters in transductive node classification on the Citeseer dataset in terms of Micro-F1.

and $p_2 = p_{r,2} = p_{m,2}$. In other words, p_1 and p_2 control the generation of the two graph views. Note that we only change these four parameters in the sensitivity analysis, other parameters remain the same as previously described.

The results on the Citeseer dataset is shown are Figure 2. From the figure, it can be observed that the performance of node classification in terms of Micro-F1 is relatively stable when the parameters are not too large, as shown in the plateau in the figure. We thus conclude that overall, our model is insensitive to these probabilities, demonstrating the robustness to hyperparameter tuning. If the probability is set too large (e.g., > 0.5), the original graph will be heavily undermined. For example, when $p_r = 0.9$, almost every existing edge has been removed, leading to isolated nodes in the generated graph views. Then, under such circumstance, the graph convolutional network is hard to learn useful information from node neighborhoods. Therefore, the learnt node embeddings in the two graph views are not distinctive enough, which will result in difficulty of optimizing the contrastive objective.

C.2 Ablation Studies

In this section, we perform ablation studies on the two schemes for generating graph views, removing edge (RE) and masking node features (MF), to verify the effectiveness of the proposed hybrid scheme. We denote GRACE (–RE) as the model without removing edges and GRACE (–MF) as the model without masking node features. We report the performance of GRACE (–RE), GRACE (–MF) and the original model GRACE on transductive node classification under the identical settings as previous, except for different enabled schemes. The results are presented in Table 5.

It is seen that our hybrid approach that jointly applies RE and MF significantly outperform two downgraded models that only use one standalone method RE or MF. These results verify the effectiveness of our proposed scheme for graph corruption, and further show the necessity of jointly considering corruption at both graph topology and node feature levels.

Table 5: The performance of model variants along with the original GRACE model in the ablation study in terms of accuracy of node classification. GRACE (–RE) and GRACE (–MF) denote the model without removing edges and masking node features respectively.

Method	Cora	Citeseer	Pubmed	DBLP
GRACE	**83.2±0.5**	**72.1±0.5**	**86.7±0.1**	**84.2±0.1**
GRACE (–RE)	82.3±0.4	72.0±0.4	84.8±0.2	83.6±0.2
GRACE (–MF)	81.6±0.4	69.9±0.6	85.7±0.1	83.5±0.1

C.3 Comparison with InfoNCE Loss

In this section, we consider another widely-used objective, the InfoNCE loss [23] , in contrastive methods. For fair comparison, we measure the node similarities between two graph views using the InfoNCE objective, which is defined as

$$\mathcal{J}_{\text{NCE}} = \frac{1}{2}\left[\ell_{\text{NCE}}(\boldsymbol{V}, \boldsymbol{U}) + \ell_{\text{NCE}}(\boldsymbol{V}, \boldsymbol{U})\right], \tag{9}$$

where the pairwise objective is defined by $\ell_{\text{NCE}}(\boldsymbol{U}, \boldsymbol{V}) \triangleq \frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{\theta(\boldsymbol{u}_i, \boldsymbol{v}_i)}}{\frac{1}{N}\sum_{j=1}^{N} e^{\theta(\boldsymbol{u}_i, \boldsymbol{v}_j)}}$. $\ell_{\text{NCE}}(\boldsymbol{V}, \boldsymbol{U})$ can be defined symmetrically. The modified model is denoted as GRACE–NCE hereafter. We report the performance of GRACE–NCE on transductive node classification under identical settings as with the original model GRACE. The results are summarized in Table 6.

From the table, we can clearly see that the performance of the variant model GRACE-NCE is inferior to that of the original model GRACE on all four datasets. The results empirically demonstrate that, although InfoNCE is a stricter estimator of the mutual information, our objective is more effective and shows better downstream performance, which is consistent with previous observations in visual representation learning [26]. We believe that the superior performance of our objective compared to InfoNCE can be attributed to the inclusion of more negative samples. Specifically, we take intra-view negative pairs into consideration in our objective, which can be viewed as a regularization against the smoothing problem brought by graph convolution operators.

Table 6: The performance of GRACE and GRACE–NCE in transductive node classification on four citation datasets.

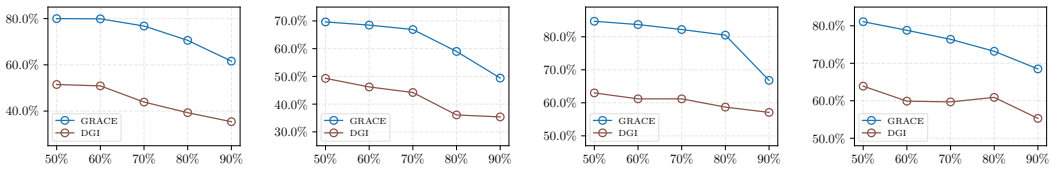
Method	Cora	Citeseer	Pubmed	DBLP
GRACE	**83.2±0.5**	**72.1±0.5**	**86.7±0.1**	**84.2±0.1**
GRACE–NCE	82.1±0.4	70.9±0.6	85.0±0.1	82.1±0.1

C.4 Robustness to Sparse Features

As discussed before, for existing work DGI, it is relatively easy to generate negative samples for nodes having dense features using the feature shuffling scheme. However, when node features are sparse, feature shuffling may not be sufficient to generate different neighborhoods for nodes, which motivates our hybrid scheme that corrupts the original graph at both topology and attribute levels.

In this section, we conduct experiments with randomly contaminating the training data by masking a certain portion of the node features to zeros. Specifically, we vary the contamination rate of node features from 0.5 to 0.9 on four citation networks. We conduct experiments on transductive node classification with all other parameters being the same as previously described. The performance in terms of accuracy is plotted in Figure 3.

From the figures, we can see that GRACE consistently outperforms DGI with large margins under different contamination rates, demonstrating the robustness of our proposed GRACE model to sparse features. We attribute the robustness of GRACE to the superiority of our proposed RE method for graph corruption at topology level, since RE is capable of constructing different topology context for nodes without dependence on node features. These results once again verify the necessity of considering graph corruption at both topology and attribute levels. Note that, when a large portion of node features are masked, e.g., 90% features are masked, both GRACE and DGI perform poorly. This may be explained from the fact that, when the node features are overly contaminated, nodes are highly sparse such that the GNN model is ineffective to extract useful information from nodes, leading to performance deterioration.



(a) Cora (b) Citeseer (c) Pubmed (d) DBLP

Figure 3: The performance of GRACE and DGI in transductive node classification in terms of Micro-F1 on four citation datasets with a portion of node features masked under different masking rates.

D Detailed Proofs

D.1 Proof of Theorem 1

Theorem 1. *Let $\mathbf{X}_i = \{\boldsymbol{x}_k\}_{k\in\mathcal{N}(i)}$ be the neighborhood of node v_i that collectively maps to its output embedding, where $\mathcal{N}(i)$ denotes the set of neighbors of node v_i specified by GNN architectures, and $\mathbf{X}$ be the corresponding random variable with a uniform distribution $p(\mathbf{X}_i) = \frac{1}{N}$. Given two random variables $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{F'}$ being the embedding in the two views, with their joint distribution denoted as $p(\mathbf{U}, \mathbf{V})$, our objective $\mathcal{J}$ is a lower bound of MI between encoder input $\mathbf{X}$ and node representations in two graph views $\mathbf{U}, \mathbf{V}$. Formally,*

$$\mathcal{J} \leq I(\mathbf{X}; \mathbf{U}, \mathbf{V}). \tag{10}$$

Proof. We first show the connection between our objective $\mathcal{J}$ and the InfoNCE objective [23] , which can be defined as [25]

$$I_{\text{NCE}}(\mathbf{U};\mathbf{V}) \triangleq \mathbb{E}_{\prod_i p(\boldsymbol{u}_i,\boldsymbol{v}_i)}\left[\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{\theta(\boldsymbol{u}_i,\boldsymbol{v}_i)}}{\frac{1}{N}\sum_{j=1}^{N}e^{\theta(\boldsymbol{u}_i,\boldsymbol{v}_j)}}\right],$$

where the critic function is defined as $\theta(\boldsymbol{x},\boldsymbol{y}) = s(g(\boldsymbol{x}),g(\boldsymbol{y}))$. We further define $\rho_r(\boldsymbol{u}_i) = \sum_{j=1}^{N}\mathbb{1}_{[i\neq j]}\exp(\theta(\boldsymbol{u}_i,\boldsymbol{u}_j)/\tau)$, $\rho_c(\boldsymbol{u}_i) = \sum_{j=1}^{N}\exp(\theta(\boldsymbol{u}_i,\boldsymbol{v}_j)/\tau)$ for convenience of notation. Note that $\rho_r(\boldsymbol{v}_i)$ and $\rho_c(\boldsymbol{v}_i)$ can be defined symmetrically. Then, our objective $\mathcal{J}$ can be rewritten as

$$\mathcal{J} = \mathbb{E}_{\prod_i p(\boldsymbol{u}_i,\boldsymbol{v}_i)}\left[\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(\theta(\boldsymbol{u}_i,\boldsymbol{v}_i)/\tau)}{\sqrt{(\rho_c(\boldsymbol{u}_i)+\rho_r(\boldsymbol{u}_i))\,(\rho_c(\boldsymbol{v}_i)+\rho_r(\boldsymbol{v}_i))}}\right]. \tag{11}$$

Using the notation of ρ_c, the InfoNCE estimator I_{NCE} can be written as

$$I_{\text{NCE}}(\mathbf{U},\mathbf{V}) = \mathbb{E}_{\prod_i p(\boldsymbol{u}_i,\boldsymbol{v}_i)}\left[\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(\theta(\boldsymbol{u}_i,\boldsymbol{v}_i)/\tau)}{\rho_c(\boldsymbol{u}_i)}\right]. \tag{12}$$

Therefore,

$$\begin{aligned}2\mathcal{J} &= I_{\text{NCE}}(\mathbf{U},\mathbf{V}) - \mathbb{E}_{\prod_i p(\mathbf{u}_i,\boldsymbol{v}_i)}\left[\frac{1}{N}\sum_{i=1}^{N}\log\left(1+\frac{\rho_r(\boldsymbol{u}_i)}{\rho_c(\boldsymbol{u}_i)}\right)\right]\\ &\quad + I_{\text{NCE}}(\mathbf{V},\mathbf{U}) - \mathbb{E}_{\prod_i p(\boldsymbol{u}_i,\boldsymbol{v}_i)}\left[\frac{1}{N}\sum_{i=1}^{N}\log\left(1+\frac{\rho_r(\boldsymbol{v}_i)}{\rho_c(\boldsymbol{v}_i)}\right)\right]\\ &\leq I_{\text{NCE}}(\mathbf{U},\mathbf{V}) + I_{\text{NCE}}(\mathbf{V},\mathbf{U}).\end{aligned} \tag{13}$$

According to [25], the InfoNCE estimator is a lower bound of the true MI, i.e.

$$I_{\text{NCE}}(\mathbf{U},\mathbf{V}) \leq I(\mathbf{U};\mathbf{V}). \tag{14}$$

Thus, we arrive at

$$2\mathcal{J} \leq I(\mathbf{U};\mathbf{V}) + I(\mathbf{V};\mathbf{U}) = 2I(\mathbf{U};\mathbf{V}), \tag{15}$$

which leads to the inequality

$$\mathcal{J} \leq I(\mathbf{U};\mathbf{V}). \tag{16}$$

According to the data processing inequality, which states that, for all random variables $\mathbf{X},\mathbf{Y},\mathbf{Z}$ satisfying the Markov relation $\mathbf{X}\to\mathbf{Y}\to\mathbf{Z}$, the inequality $I(\mathbf{X};\mathbf{Z}) \leq I(\mathbf{X};\mathbf{Y})$ holds. Then, we observe that $\mathbf{X},\mathbf{U},\mathbf{V}$ satisfy the relation $\mathbf{U}\leftarrow\mathbf{X}\to\mathbf{V}$. Since, $\mathbf{U}$ and $\mathbf{V}$ are conditionally independent after observing $\mathbf{X}$, the relation is Markov equivalent to $\mathbf{U}\to\mathbf{X}\to\mathbf{V}$, which leads to $I(\mathbf{U};\mathbf{V}) \leq I(\mathbf{U};\mathbf{X})$. We further notice that the relation $\mathbf{X}\to(\mathbf{U},\mathbf{V})\to\mathbf{U}$ holds, and hence it follows that $I(\mathbf{X};\mathbf{U}) \leq I(\mathbf{X};\mathbf{U},\mathbf{V})$. Combining the two inequalities yields the required inequality

$$I(\mathbf{U};\mathbf{V}) \leq I(\mathbf{X};\mathbf{U},\mathbf{V}). \tag{17}$$

Following Eq. (16) and Eq. (17), we finally arrive at inequality

$$\mathcal{J} \leq I(\mathbf{X};\mathbf{U},\mathbf{V}), \tag{18}$$

which concludes the proof. □

D.2 Proof of Theorem 2

Theorem 2. *When the projection function g is the identity function and we measure embedding similarity by simply taking inner product, i.e. $s(\boldsymbol{u},\boldsymbol{v}) = \boldsymbol{u}^\top\boldsymbol{v}$, and further assuming that positive pairs are far more aligned than negative pairs, minimizing the pairwise objective $\ell(\boldsymbol{u}_i,\boldsymbol{v}_i)$ coincides with maximizing the triplet loss, as given in the sequel*

$$-\ell(\boldsymbol{u}_i,\boldsymbol{v}_i) \propto 4N\tau + \sum_{j=1}^{N}\mathbb{1}_{[j\neq i]}\left[\left(\|\boldsymbol{u}_i-\boldsymbol{v}_i\|^2 - \|\boldsymbol{u}_i-\boldsymbol{v}_j\|^2\right) + \left(\|\boldsymbol{u}_i-\boldsymbol{v}_i\|^2 - \|\boldsymbol{u}_i-\boldsymbol{u}_j\|^2\right)\right]. \tag{19}$$

Proof. Based on the assumptions, we can rearrange the pairwise objective as

$$
\begin{aligned}
-\ell(\boldsymbol{u}_i, \boldsymbol{v}_i) &= -\log \frac{\exp\left(\boldsymbol{u}_i^\top \boldsymbol{v}_i/\tau\right)}{\sum_{k=1}^N \exp\left(\boldsymbol{u}_i^\top \boldsymbol{v}_k/\tau\right) + \sum_{k=1}^N \mathbb{1}_{[k\neq i]} \exp\left(\boldsymbol{u}_i^\top \boldsymbol{u}_k/\tau\right)} \\
&= \log\left(1 + \sum_{k=1}^N \mathbb{1}_{[k\neq i]} \exp\left(\frac{\boldsymbol{u}_i^\top \boldsymbol{v}_k - \boldsymbol{u}_i^\top \boldsymbol{v}_i}{\tau}\right) + \sum_{k=1}^N \mathbb{1}_{[k\neq i]} \exp\left(\frac{\boldsymbol{u}_i^\top \boldsymbol{u}_k - \boldsymbol{u}_i^\top \boldsymbol{v}_i}{\tau}\right)\right).
\end{aligned} \tag{20}
$$

By Taylor expansion of first order,

$$
\begin{aligned}
-\ell(\boldsymbol{u}_i, \boldsymbol{v}_i) &\approx \sum_{k=1}^N \mathbb{1}_{[k\neq i]} \exp\left(\frac{\boldsymbol{u}_i^\top \boldsymbol{v}_k - \boldsymbol{u}_i^\top \boldsymbol{v}_i}{\tau}\right) + \sum_{k=1}^N \mathbb{1}_{[k\neq i]} \exp\left(\frac{\boldsymbol{u}_i^\top \boldsymbol{u}_k - \boldsymbol{u}_i^\top \boldsymbol{v}_i}{\tau}\right) \\
&\approx 2 + \frac{1}{\tau}\left[\sum_{k=1}^N \mathbb{1}_{[k\neq i]} (\boldsymbol{u}_i^\top \boldsymbol{v}_k - \boldsymbol{u}_i^\top \boldsymbol{v}_i) + \sum_{k=1}^N \mathbb{1}_{[k\neq i]} (\boldsymbol{u}_i^\top \boldsymbol{u}_k - \boldsymbol{u}_i^\top \boldsymbol{v}_i)\right] \\
&= 2 - \frac{1}{2\tau} \sum_{k=1}^N \mathbb{1}_{[k\neq i]} \left[\left(\|\boldsymbol{u}_i - \boldsymbol{v}_k\|^2 - \|\boldsymbol{u}_i - \boldsymbol{v}_i\|^2\right) + \left(\|\boldsymbol{u}_i - \boldsymbol{u}_k\|^2 - \|\boldsymbol{u}_i - \boldsymbol{v}_i\|^2\right)\right] \\
&\propto 4N\tau + \sum_{k=1}^N \mathbb{1}_{[k\neq i]} \left[\left(\|\boldsymbol{u}_i - \boldsymbol{v}_i\|^2 - \|\boldsymbol{u}_i - \boldsymbol{v}_k\|^2\right) + \left(\|\boldsymbol{u}_i - \boldsymbol{v}_i\|^2 - \|\boldsymbol{u}_i - \boldsymbol{u}_k\|^2\right)\right],
\end{aligned} \tag{21}
$$

which concludes the proof. □

References

[1] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. DeepWalk: Online Learning of Social Representations. In *KDD*, 2014.

[2] Aditya Grover and Jure Leskovec. node2vec: Scalable Feature Learning for Networks. In *KDD*, 2016.

[3] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. In *NIPS*, 2013.

[4] Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. Network Embedding as Matrix Factorization: Unifying DeepWalk, LINE, PTE, and node2vec. In *WSDM*, 2018.

[5] Leonardo Filipe Rodrigues Ribeiro, Pedro H. P. Saverese, and Daniel R. Figueiredo. struc2vec: Learning Node Representations from Structural Identity. In *KDD*, 2017.

[6] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*, 2017.

[7] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. In *ICLR*, 2018.

[8] Fenyu Hu, Yanqiao Zhu, Shu Wu, Liang Wang, and Tieniu Tan. Hierarchical Graph Convolutional Networks for Semi-supervised Node Classification. In *IJCAI*, 2019.

[9] Thomas N. Kipf and Max Welling. Variational Graph Auto-Encoders. In *BDL@NIPS*, 2016.

[10] William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive Representation Learning on Large Graphs. In *NIPS*, 2017.

[11] Ralph Linsker. Self-Organization in a Perceptual Network. *IEEE Computer*, 1988.

[12] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised Feature Learning via Non-Parametric Instance Discrimination. In *CVPR*, 2018.

[13] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive Multiview Coding. *arXiv.org*, June 2019.

[14] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum Contrast for Unsupervised Visual Representation Learning. In *CVPR*, 2020.

[15] Philip Bachman, R. Devon Hjelm, and William Buchwalter. Learning Representations by Maximizing Mutual Information Across Views. In *NeurIPS*, 2019.

[16] Mang Ye, Xu Zhang, Pong C. Yuen, and Shih-Fu Chang. Unsupervised Embedding Learning via Invariant and Spreading Instance Feature. In *CVPR*, 2019.

[17] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A Simple Framework for Contrastive Learning of Visual Representations. *arXiv.org*, February 2020.

[18] Petar Veličković, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R. Devon Hjelm. Deep Graph Infomax. In *ICLR*, 2019.

[19] Suzanna Becker and Geoffrey E. Hinton. Self-Organizing Neural Network That Discovers Surfaces in Random-Dot Stereograms. *Nature*, 355(6356), 1992.

[20] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised Representation Learning by Predicting Image Rotations. In *ICLR*, 2018.

[21] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Colorization as a Proxy Task for Visual Understanding. In *CVPR*, 2017.

[22] R. Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Philip Bachman, Adam Trischler, and Yoshua Bengio. Learning Deep Representations by Mutual Information Estimation and Maximization. In *ICLR*, 2019.

[23] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning with Contrastive Predictive Coding. *arXiv.org*, 2018.

[24] Olivier J. Hénaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, S. M. Ali Eslami, and Aaron van den Oord. Data-Efficient Image Recognition with Contrastive Predictive Coding. *arXiv.org*, May 2019.

[25] Ben Poole, Sherjil Ozair, Aäron van den Oord, Alexander A. Alemi, and George Tucker. On Variational Bounds of Mutual Information. In *ICML*, 2019.

[26] Michael Tschannen, Josip Djolonga, Paul K. Rubenstein, Sylvain Gelly, and Mario Lucic. On Mutual Information Maximization for Representation Learning. In *ICLR*, 2020.

[27] William L. Hamilton, Rex Ying, and Jure Leskovec. Representation Learning on Graphs: Methods and Applications. *IEEE Data Eng. Bull.*, 2017.

[28] Michael Gutmann and Aapo Hyvärinen. Noise-Contrastive Estimation of Unnormalized Statistical Models, with Applications to Natural Image Statistics. *JMLR*, 2012.

[29] Felix Wu, Tianyi Zhang, Amauri Holanda de Souza Jr., Christopher Fifty, Tao Yu, and Kilian Q. Weinberger. Simplifying Graph Convolutional Networks. In *ICML*, 2019.

[30] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks From Overfitting. *JMLR*, 15(1), 2014.

[31] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. DropEdge: Towards Deep Graph Convolutional Networks on Node Classification. In *ICLR*, 2020.

[32] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A Unified Embedding for Face Recognition and Clustering. In *CVPR*, 2015.

[33] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. Collective Classification in Network Data. *AI Magazine*, 2008.

[34] Aleksandar Bojchevski and Stephan Günnemann. Deep Gaussian Embedding of Graphs: Unsupervised Inductive Learning via Ranking. In *ICLR*, 2018.

[35] Marinka Zitnik and Jure Leskovec. Predicting Multicellular Function Through Multi-layer Tissue Networks. *Bioinform.*, 2017.

[36] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016.

[37] Jie Chen, Tengfei Ma, and Cao Xiao. FastGCN: Fast Learning with Graph Convolutional Networks via Importance Sampling. In *ICLR*, 2018.

[38] Jiani Zhang, Xingjian Shi, Junyuan Xie, Hao Ma, Irwin King, and Dit-Yan Yeung. GaAN: Gated Attention Networks for Learning on Large and Spatiotemporal Graphs. In *UAI*, 2018.

[39] Zhilin Yang, William W. Cohen, and Ruslan R. Salakhutdinov. Revisiting Semi-Supervised Learning with Graph Embeddings. In *ICML*, 2016.

[40] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. GloVe: Global Vectors for Word Representation. In *EMNLP*, 2014.

[41] Matthias Fey and Jan Eric Lenssen. Fast Graph Representation Learning with PyTorch Geometric. In *RLGM@ICLR*, 2019.

[42] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *NeurIPS*, 2019.

[43] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake VanderPlas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. Scikit-learn: Machine Learning in Python. *JMLR*, 2011.

[44] Xavier Glorot and Yoshua Bengio. Understanding the Difficulty of Training Deep Feedforward Neural Networks. In *AISTATS*, 2010.

[45] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *ICLR*, 2015.