

# Improving Graph Representation Learning by Contrastive Regularization

Kaili Ma, Haochen Yang, Han Yang, Tatiana Jin, Pengfei Chen, Yongqiang Chen, Barakeel Fanseu Kamhoua, James Cheng

{klma,hyang,tjin,pfchen,yqchen,kamhoua,jcheng}@cse.cuhk.edu.hk, yanghaochengary@gmail.com

Department of Computer Science and Engineering, The Chinese University of Hong Kong

## ABSTRACT

Graph representation learning is an important task with applications in various areas such as online social networks, e-commerce networks, WWW and semantic webs. For unsupervised graph representation learning, many algorithms such as Node2Vec and GraphSAGE make use of “negative sampling” and/or noise contrastive estimation loss. This bears similar ideas to contrastive learning, which “contrasts” the node representation similarities of semantically similar (positive) pairs against those of negative pairs. However, despite the success of contrastive learning, we found that directly applying this technique to graph representation learning models (e.g., graph convolutional networks) does not always work. We theoretically analyze the generalization performance and propose a light-weight regularization term that avoids the high scales of node representations’ norms and the high variance among them to improve the generalization performance. Our experimental results further validate that this regularization term significantly improves the representation quality across different node similarity definitions and outperforms the state-of-the-art methods.

## KEYWORDS

Graph representation, Contrastive learning, Regularization

## 1 INTRODUCTION

Graph is widely used to capture rich information (e.g., hierarchical structures, communities) in data from various domains such as social networks, e-commerce networks, knowledge graphs, WWW and semantic webs. By incorporating graph topology and node/edge features into machine learning models, *graph representation learning* has achieved great success in many important applications such as node classification, link prediction, and graph clustering.

A large number of graph representation learning algorithms [1, 5, 7, 16, 18, 25, 32, 33, 35, 36, 41, 45, 46, 50] have been proposed. Among them, many [16, 18, 33, 41] are designed in an unsupervised manner and make use of “*negative sampling*” to learn node representations. This design shares similar ideas as *contrastive learning* [2, 3, 6, 9, 19–21, 28, 42–44, 48], which “contrasts” the similarities of the representations of similar (or positive) node pairs against those of negative pairs. These algorithms adopt *noise contrastive estimation loss (NCEloss)*, while they differ in the definition of node similarity (hence the design of contrastive pairs) and encoder design.

Existing graph representation learning algorithms mainly fall into three categories: *adjacency matrix factorization based models* [1, 5, 35], *skip-gram based models* [7, 16, 33], and *graph neural networks (GNNs)* [18, 25, 36, 45, 50]. We focus on GNNs as GNN models can not only capture graph topology information as the skip-gram and

factorization based models, but also incorporate node/edge features. Specifically, we formulate a **contrastive GNN framework** with four components: *a similarity definition*, *a GNN encoder*, *a contrastive loss function*, and possibly *a downstream classification task*.

While graph representation algorithms using the “negative sampling” approach are shown to achieve good performance empirically, there is a lack of theoretical analysis on the generalization performance. In addition, we also found that directly applying contrastive pairs and NCEloss to existing GNN models, e.g., graph convolutional networks (GCNs) [25], does not always work well (as shown in Section 7). In order to understand the generalization performance of the algorithms and find out when the direct application of contrastive pairs and NCEloss does not work, we derive a generalization bound for our contrastive GNN framework using the theoretical framework proposed in [38]. Our generalization bound reveals that *the high scales of node representations’ norms and the high variance among them are two main factors that hurt the generalization performance*.

To solve the problems caused by the two factors, we propose a novel regularization method, **Contrast-Reg**. Contrast-Reg uses a regularization vector  $r$ , which is a random vector with each element in the range (0, 1]. We learn a graph representation model by forcing the representations of all nodes to be similar to  $r$  and all the contrastive representations calculated by shuffling node features to be dissimilar to  $r$ . We show from the geometric perspective that Contrast-Reg stabilizes the scales of norms and reduces their variance. We also validate by experiments that Contrast-Reg significantly improves the quality of node representations for a popular GNN model using different similarity definitions.

**Outline.** Section 2 discusses related work. Section 3 gives some preliminaries and Section 4 analyzes the generalization bound. Section 5 proposes Contrast-Reg and Section 6 presents the contrastive GNN framework. Section 7 reports the experimental results.

## 2 RELATED WORK

**Graph representation learning.** Many graph representation learning models have been proposed. Factorization based models [1, 5, 35] factorize an adjacency matrix to obtain node representations. Random walk based models such as DeepWalk [33] sample node sequences as the input to skip-gram models to compute the representation for each node. Node2vec [16] balances depth-first and breadth-first random walk when it samples node sequences. HARP [7] compresses nodes into super-nodes to obtain a hierarchical graph to provide hierarchical information to random walk. GNN models [18, 25, 36, 45, 50] have shown great capability in capturing both graph topology and node/edge feature information. Most GNN models follow a neighborhood aggregation schema, in which each

node receives and aggregates the information from its neighbors in each GNN layer, i.e., for the  $k$ -th layer,  $\tilde{h}_i^k = \text{aggregate}(\tilde{h}_j^{k-1}, j \in \text{neighborhood}(i))$ , and  $h_i^k = \text{combine}(\tilde{h}_i^k, h_i^{k-1})$ . This work proposes a regularization method, Contrast-Reg, for GNN models and improves their performance for downstream tasks.

**Contrastive learning.** Contrastive learning is a self-supervised learning method that learns representations by contrasting positive pairs against negative pairs. Contrastive pairs can be constructed in different ways for different types of data and tasks, such as multi-view [42, 43], target-to-noise [20, 44], mutual information [3, 21], instance discrimination [48], context co-occurrence [28], clustering [2, 6], and multiple data augmentation [9]. In addition to the above unsupervised learning settings, [9, 42, 44] also show great success in capturing information that can be transferred to new tasks from different domains.

Contrastive learning has been successfully applied in many graph representation learning models such as [16, 18, 32, 33, 41, 46]. In this work, we apply contrastive learning in GNNs and propose a contrastive GNN framework. The state-of-the-art models such as [32, 46], which both use the GCN encoder, can be seen as special instances of this algorithmic framework. We show that with the same GCN encoder and similar contrastive pair designs, our models can significantly outperform [32, 46] by adopting Contrast-Reg.

**Noise Contrastive Estimation loss.** NCEloss was originally proposed to reduce the computation cost of estimating the parameters of a probabilistic model using logistic regression to discriminate between the observed data and artificially generated noises [13, 17, 29]. It has been successfully applied to contrastive learning [3, 9, 20, 21, 42–44, 48]. There are works aiming to explain the success of NCEloss. Gutmann and Hyvärinen [17] proved that when NCEloss serves as the objective function of the parametric density estimation problem, the estimation of the parameters converges in probability to the optimal estimation. Yang et al. [54] showed that when NCEloss is applied in graph representation learning, the mean squared error of the similarity between two nodes is related to the negative sampling strategy. However, the definition of optimal representations or optimal parameters does not consider downstream tasks, but is based on pre-defined structures. In this paper, we adopt the theoretical settings from the contrastive learning framework proposed by Saunshi et al. [38] to analyze the generalizability of NCEloss in downstream tasks, i.e., linear classification tasks. To the best of our knowledge, we are the first to theoretically analyze NCEloss under the contrastive learning setting [38].

**Node-level similarity.** NCEloss has been adopted in many graph representation learning models to capture different types of node-level similarity. We characterize them as follows:

- **Structural similarity:** We may capture structural similarity from different angles. From the graph theory perspective, Graph-Wave [12] leverages the diffusion of spectral graph wavelets to capture structural similarity, and struc2vec [37] uses a hierarchy to measure node similarity at different scales. From the induced subgraph perspective, GCC [34] treats the induced subgraphs of the same ego network as similar pairs and those from different ego networks as dissimilar pairs. To capture the community structure, vGraph [40] utilizes the high correlation of community detection and node representations to make node representations

contain more community structure information. To capture the global-local structure, DGI [46] maximizes the mutual information between node representations and graph representations to allow node representations to contain more global information.

- **Attribute similarity:** Nodes with similar attributes are likely to have similar representations. GMI [32] maximizes the mutual information between node attributes and high-level representations, and Hu et al. [23] applies attribute masking to help capture domain-specific knowledge.
- **Proximity similarity:** Most random walk based models such as DeepWalk [33], Node2vec [16], and LINE [41] share an assumption that nodes with more proximity have higher probability to share the same label.

We will show that Contrast-Reg facilitates the contrastive training of graph representation learning models regardless of the different designs of contrastive pairs being used, and thus is helpful in capturing all types of similarities.

**Regularization for graph representation learning.** In addition to the general regularization terms used in machine learning such as L1/L2 regularization, there are regularizers proposed for graph representation learning models. GraphAT [14] and BVAT [10] add adversarial perturbation  $\frac{\partial f}{\partial x}$  to the input data  $x$  as a regularizer to obtain more robust models. GraphSGAN [11] generates fake input data in low density region by taking a generative adversarial network as regularizer. P-reg [52] makes use of the smoothness property in real-world graphs to improve GNN models. Most of the above regularizers are designed for supervised tasks and perform well only in supervised settings, while Contrast-Reg is the first regularizer designed for contrastive learning and achieves excellent performance in unsupervised settings. We will also show its advantages over traditional regularizers, e.g., weight decay (L2 regularization of the model parameters) [26] in Section 5.

### 3 PRELIMINARIES

We briefly discuss the theoretical framework [38] for contrastive learning and NCEloss, which is the foundation of Section 4.

#### 3.1 Concepts in Contrastive Learning

Consider the feature space  $\mathcal{X}$ , the goal of contrastive learning is to train an encoder  $f \in \mathcal{F} : \mathcal{X} \rightarrow \mathbb{R}^d$  for all input data points  $x \in \mathcal{X}$  by constructing positive pairs  $(x, x^+)$  and negative pairs  $(x, x_1^-, \dots, x_K^-)$ . To formally analyze the behavior of contrastive learning, Saunshi et al. [38] introduce the following concepts.

- **Latent classes:** Data are considered as drawn from latent classes  $C$  with distribution  $\rho$ . Further, distribution  $\mathcal{D}_c$  is defined over feature space  $\mathcal{X}$  that is associated with a class  $c \in C$  to measure the relevance between  $x$  and  $c$ .
- **Semantic similarity:** Positive samples are drawn from the same latent classes, with distribution

$$\mathcal{D}_{\text{sim}}(x, x^+) = \mathbb{E}_{c \in \rho} [\mathcal{D}_c(x)\mathcal{D}_c(x^+)], \quad (1)$$

while negative samples are drawn randomly from all possible data points, i.e., the marginal of  $\mathcal{D}_{\text{sim}}$ , as

$$\mathcal{D}_{\text{neg}}(x^-) = \mathbb{E}_{c \in \rho} [\mathcal{D}_c(x^-)] \quad (2)$$

- *Supervised tasks:* Denote  $K$  as the number of negative samples. The object of the supervised task, i.e., feature-label pair  $(x, c)$ , is sampled from

$$\mathcal{D}_{\mathcal{T}}(x, c) = \mathcal{D}_c(x)\mathcal{D}_{\mathcal{T}}(c),$$

where  $\mathcal{D}_{\mathcal{T}}(c) = \rho(c|c \in \mathcal{T})$ , and  $\mathcal{T} \subseteq C$  with  $|\mathcal{T}| = K + 1$ . Mean classifier  $W^\mu$  is naturally imposed to bridge the gap between the representation learning performance and linear separability of learn representations, as

$$W_c^\mu := \mu_c = \mathbb{E}_{x \sim \mathcal{D}_c}[f(x)].$$

- *Empirical Rademacher complexity:* Suppose  $\mathcal{F} : \mathcal{X} \rightarrow [1, 0]$ . Given a sample  $\mathcal{S}$ ,

$$\mathcal{R}_{\mathcal{S}}(\mathcal{F}) = \mathbb{E}_{\vec{e}} \left[ \sup_{f \in \mathcal{F}} \vec{e}^T f(\mathcal{S}) \right],$$

where  $\vec{e} = (e_1, \dots, e_m)^T$ , with  $e_i$  are independent random variables taking values uniformly from  $\{-1, +1\}$ .

In addition, the theoretical framework in [38] makes an assumption: encoder  $f$  is bounded, i.e.,  $\max_{x \in \mathcal{X}} \|f(x)\| \leq R^2$ ,  $R \in \mathbb{R}$ .

### 3.2 Contrastive Learning with NCEloss

The contrastive loss defined by Saunshi et al. [38] is

$$\mathcal{L}_{un} := \mathbb{E}_{\substack{(x, x^+) \sim \mathcal{D}_{sim}, \\ (x_1^-, \dots, x_K^-) \sim \mathcal{D}_{neg}}} [\ell(\{f(x)^T(f(x^+) - f(x_i^-))\}_{i=1}^K)],$$

where  $\ell$  can be the hinge loss as  $\ell(v) = \max\{0, 1 + \max_i \{-v_i\}\}$  or the logistic loss as  $\ell(v) = \log_2(1 + \sum_i \exp(-v_i))$ . And its supervised counterpart is defined as

$$\mathcal{L}_{sup}^\mu := \mathbb{E}_{(x, c) \sim \mathcal{D}_{\mathcal{T}}(x, c)} [\ell(\{f(x)^T \mu_c - f(x)^T \mu_{c'}\}_{c' \neq c})].$$

A more powerful loss function, NCEloss, used in [13, 29, 46, 54], can be framed as

$$\begin{aligned} \mathcal{L}_{nce} &:= \\ &- \mathbb{E}_{\substack{(x, x^+) \sim \mathcal{D}_{sim}, \\ (x_1^-, \dots, x_K^-) \sim \mathcal{D}_{neg}}} \left[ \log \sigma(f(x)^T f(x^+)) + \sum_{k=1}^K \log \sigma(-f(x)^T f(x_k^-)) \right], \end{aligned} \quad (3)$$

and its empirical counterpart with  $M$  samples  $(x_i, x_i^+, x_{i1}^-, \dots, x_{iK}^-)_{i=1}^M$  is given as

$$\hat{\mathcal{L}}_{nce} := -\frac{1}{M} \sum_{i=1}^M \left[ \log \sigma(f(x_i)^T f(x_i^+)) + \sum_{k=1}^K \log \sigma(-f(x_i)^T f(x_{ik}^-)) \right], \quad (4)$$

where  $\sigma(\cdot)$  is the sigmoid function.

For its supervised counterpart, it is exactly the cross entropy loss for the  $(K + 1)$ -way multi-class classification task:

$$\mathcal{L}_{sup}^\mu := -\mathbb{E}_{(x, c) \sim \mathcal{D}_{\mathcal{T}}(x, c)} [\log \sigma(f(x)^T \mu_c) + \log \sigma(-f(x)^T \mu_{c'}) | c' \neq c]. \quad (5)$$

## 4 THEORETICAL ANALYSES ON NCELOSS

In this section, we first give the upper bound of supervised loss when training a model  $f$  using NCEloss. Then we discuss the generalization bound of NCEloss along with the generalization bounds of hinge loss and logistic loss, and show their limitations.

### 4.1 The Generalization Bound of NCEloss

We give the generalization error of function class  $\mathcal{F}$  on the unsupervised loss function  $\mathcal{L}_{nce}$  in Theorem 4.1. Since we focus the regularization in contrastive learning, we give the result based on a single negative sample, i.e.,  $K = 1$ .

Let  $c, c'$  be two classes sampled independently from latent classes  $C$  with distribution  $\rho$ . Let  $\tau = \mathbb{E}_{c, c' \sim \rho^2} \mathbb{I}\{c = c'\}$  be the probability that  $c$  and  $c'$  come from the same class. And  $\mathcal{L}_{nce}^=(f)$  and  $\mathcal{L}_{nce}^≠(f)$  are NCEloss when negative samples come from the same and different class, respectively. We have the following theorem.

**THEOREM 4.1.** *For all  $f \in \mathcal{F}$ , with probability at least  $1 - \delta$ ,*

$$\mathcal{L}_{sup}^\mu(\hat{f}) \leq \mathcal{L}_{nce}^≠(f) + \beta s(f) + \eta G_{\mathcal{M}} + \alpha, \quad (6)$$

where  $\beta = \frac{\tau}{1-\tau}$ ,  $\eta = \frac{1}{1-\tau}$ ,  $\alpha = \frac{1}{1-\tau}(2 \log c' - 1)$ ,  $c' \in [1 + e^{-R^2}, 2]$ ,  $s(f) = 4\sqrt{\mathbb{E}_{(x_i, x_j) \sim \mathcal{D}_{sim}(x_i, x_j)} [(f(x_i)^T f(x_j))^2]}$ , and

$$\begin{aligned} G_{\mathcal{M}} &= \frac{8R\mathcal{R}_{\mathcal{S}}(\mathcal{F})}{M} - 8 \log(\sigma(-R^2)) \sqrt{\frac{\log \frac{4}{\delta}}{2M}} \\ &= O\left(R \frac{\mathcal{R}_{\mathcal{S}}(\mathcal{F})}{M} + R^2 \sqrt{\frac{\log \frac{1}{\delta}}{M}}\right). \end{aligned}$$

**REMARK.** *The above theorem tells us if contrastive learning algorithms with NCEloss could make  $\beta s(f) + \eta G_{\mathcal{M}} + \alpha$  converge to 0 as  $M$  increases, the picked encoder  $\hat{f} = \arg \min_{f \in \mathcal{F}} \hat{\mathcal{L}}_{nce}$  will have good performance in downstream tasks. In other words, we could guarantee contrastive learning algorithms to obtain high quality representations by minimizing  $\hat{\mathcal{L}}_{nce}$  under the condition that  $\beta s(f) + \eta G_{\mathcal{M}} + \alpha$  will converge to 0 with a large amount of data.*

To prove Theorem 4.1, we first list some key lemmas.

**LEMMA 4.2.** *For all  $f \in \mathcal{F}$ ,*

$$\mathcal{L}_{sup}^\mu(f) \leq \frac{1}{1-\tau} (\mathcal{L}_{nce}(f) - \tau). \quad (7)$$

This bound connects contrastive representation learning algorithms and its supervised counterpart. This lemma is achieved by Jensen's inequality. The details are given in Appendix A.1.

**LEMMA 4.3.** *With probability at least  $1 - \delta$  over the set  $\mathcal{S}$ , for all  $f \in \mathcal{F}$ ,*

$$\mathcal{L}_{nce}(\hat{f}) \leq \mathcal{L}_{nce}(f) + G_{\mathcal{M}}. \quad (8)$$

This bound guarantees that the chosen  $\hat{f} = \arg \min_{f \in \mathcal{F}} \mathcal{L}_{sup}^\mu(f)$  cannot be too much worse than  $f^* = \arg \min_{f \in \mathcal{F}} \mathcal{L}_{nce}$ . The proof applies Rademacher complexity of the function class [30] and vector-contraction inequality [27]. More details are given in Appendix A.2.

**LEMMA 4.4.**  $\mathcal{L}_{nce}^=(f) \leq 4s(f) + 2 \log c'$ .

This bound is derived by the loss caused by both positive and negative pairs that come from the same class, i.e., class collision. The proof uses Bernoulli's inequality (details in Appendix A.3).

**PROOF TO THEOREM 4.1.** Combining Lemma 4.2 and Lemma 4.3, we obtain with probability at least  $1 - \delta$  over the set  $\mathcal{S}$ , for all  $f \in \mathcal{F}$ ,

$$\mathcal{L}_{sup}^\mu(\hat{f}) \leq \frac{1}{1-\tau} (\mathcal{L}_{nce}(f) + G_{\mathcal{M}} - \tau) \quad (9)$$

Then, we decompose  $\mathcal{L}_{nce} = \tau \mathcal{L}_{nce}^=(f) + (1 - \tau) \mathcal{L}_{nce}^≠(f)$ , apply Lemma 4.4 to Eq. (9), and obtain the result of Theorem 4.1  $\square$

## 4.2 Discussion on the Generalization Bound

We now discuss the generalization error of NCEloss in the contrastive learning setting.

**4.2.1 Discussion on  $\text{Gen}_M$  and  $s(f)$ .**  $\text{Gen}_M$  in Eq. (6) is the generalization error in terms of *Rademacher complexity*. It shows that when the encoder function is bounded and the number of samples  $M$  is large enough,  $\hat{f}$  obtained by minimizing  $\hat{\mathcal{L}}_{\text{nce}}$  provides performance guarantee. Note that when satisfying  $\ell$  is a bounded Lipschitz continuous function and encoder  $f$  is bounded, the generalization error of different contrastive loss terms will only differ in the contraction rate, i.e., Lipschitz continuous constant.

$s(f)$  in Eq. (6) can be further rewritten as

$$\begin{aligned} s(f) &= 4\sqrt{\mathbb{E}_{(x_i, x_j) \sim \mathcal{D}_{\text{sim}}(x_i, x_j)} [f(x_i)^T f(x_j) f(x_j)^T f(x_i)]} \\ &= 4\sqrt{\mathbb{E}_{c \sim \rho} [\mathbb{E}_{x_i \sim \mathcal{D}_c} [f(x_i)^T \mathbb{E}_{x_j \sim \mathcal{D}_c} [f(x_j) f(x_j)^T] f(x_i)]]} \\ &\leq 4\sqrt{\mathbb{E}_{c \sim \rho} \left[ \max_{x \sim \mathcal{D}_c} \|f(x)\|^2 \times \|M(f, c)\|_2 \right]}, \end{aligned}$$

where  $M(f, c) := \mathbb{E}_{x \sim \mathcal{D}_c} [f(x) f(x)^T]$ . It shows that NCEloss is prone to be disturbed by large representation norms.

**4.2.2 Cases that make contrastive learning suboptimal.** There are two cases where contrastive learning algorithms cannot guarantee that  $\hat{f}$  works in downstream tasks as pointed out in [38], which also applies for NCELoss. *Case 1.* The optimal  $f$  for the downstream task can have large  $\mathcal{L}_{\text{un}}^\#$  ( $\mathcal{L}_{\text{nce}}^\#$ ) and thus failure of the algorithm, because of large spurious components in the representations that are orthogonal to the separation plain in the downstream task. *Case 2.* High intraclass deviation makes  $\mathcal{L}_{\text{un}}^\#$  ( $\mathcal{L}_{\text{nce}}^\#$ ) large even if both of its supervised counterpart losses  $\mathcal{L}_{\text{sup}}^\#$  and  $\mathcal{L}_{\text{un}}^\#$  ( $\mathcal{L}_{\text{nce}}^\#$ ) are small, resulting in failure of the algorithm.

There is an additional case for NCEloss (*Case 3*). The optimal  $f$  for the downstream task can have large  $\mathbb{E}[\|f(x)\|]$  and  $\text{Var}(\|f(x)\|)$ , which lead to large  $\mathcal{L}_{\text{nce}}^\#$  and large  $s(f)$ , even if  $f$  gives low intra-class deviation.

*Example.* Figure 1 depicts an example with  $\mathcal{F} = \{f_1, f_2\}$ ,  $C = \{c_1, c_2\}$ ,  $X = \{x_1, x_2\}$ , and  $\mathcal{D}_{c_1}(x_1) = 1$  and  $\mathcal{D}_{c_2}(x_2) = 1$ . In this example, the linear separability of  $f_1$  is better than  $f_2$  in both Figure 1a and 1b, while  $\|f_1(x_1)\| \gg \|f_1(x_2)\|$ . In the case of  $f_1(x_1)^T f_1(x_2) > 0$  (case 1 and 3), the contrastive learning algorithm using NCEloss will converge to pick  $f_2$  since  $\mathcal{L}_{\text{nce}}^\#(f_1) \gg \mathcal{L}_{\text{nce}}^\#(f_2)$  and  $s(f_1) \gg s(f_2)$ . When  $f_1(x_1)^T f_1(x_2) < 0$ ,  $f_2$  will be chosen, since  $s(f_1) \gg s(f_2)$  in Eq.(6) (case 3).

We remark that once we avoid the problems of case 3, the problems of case 1 and case 2 cannot be serious. For case 1, since when both  $\text{Var}(\|f(x)\|)$  and  $\mathbb{E}[\|f(x)\|]$  are not large, the length of the orthogonal project is comparable to the separation plane so as to avoid destroying the contrast ability. For case 2, mild scale  $\text{Var}(\|f(x)\|)$  and  $\mathbb{E}[\|f(x)\|]$  avoids large intra-class deviation caused by large variance of the representation norm.

We further show that case 3 is not an artificial case, but exists in practice. We use the training status of only using one contrastive loss computed by structural similarity on the Cora dataset [53] (Section 6) to demonstrate the issues with high expectation and

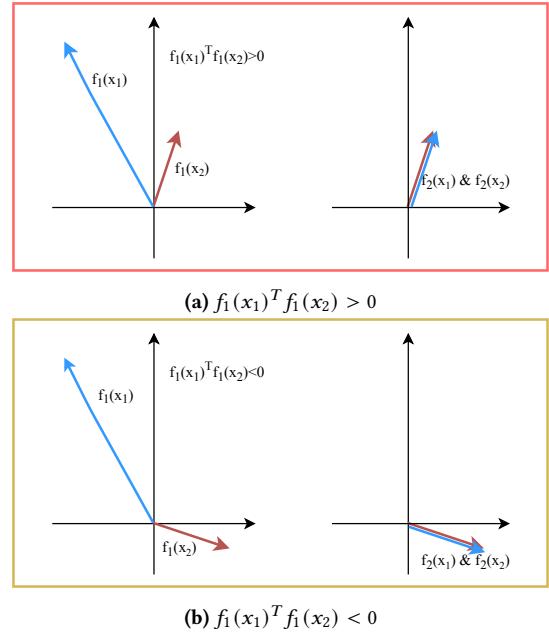


Figure 1: Cases of contrastive learning being suboptimal

high variance of representation norms. Denote

$$\begin{aligned} \mu^+ &:= \mathbb{E}_{c \sim \rho} \left[ \mathbb{E}_{(x_i, x_j) \sim \mathcal{D}_{\text{sim}}} [|f(x_i)^T f(x_j)|] \right] \\ \mu^- &:= \mathbb{E}_{c_1 \neq c_2, c_1, c_2 \sim \rho} \left[ \mathbb{E}_{x_i \sim \mathcal{D}_{c_1}, x_j \sim \mathcal{D}_{c_2}} [|f(x_i)^T f(x_j)|] \right] \end{aligned}$$

Figure 2 shows the variance and the mean of representation norms (left top), the ratio of  $\mu^+$  to  $\mu^-$  (right top),  $\mu^+$  and  $\mu^-$  (left bottom), and the testing accuracy (in a node classification task) during training for 300 epochs. The variance and mean value of representation norms increase with the progression of epochs. This increases  $\mu^+$  and  $\mu^-$  significantly, while the ratio  $\frac{\mu^+}{\mu^-}$  and representation quality (indicated by the test accuracy) decrease. In the following sections, we use the ratio  $\frac{\mu^+}{\mu^-}$  to measure the contrasting ability of models, i.e., the ability to contrast different classes.

## 5 CONTRASTIVE REGULARIZATION

The theoretical analysis in Section 4 shows that a good contrastive representation learning algorithm should satisfy the following conditions: 1. avoiding large representation norm  $\mathbb{E}[\|f(x)\|]$ ; 2. avoiding large norm variance  $\text{Var}(\|f(x)\|)$ ; 3. preserving contrast.

We remark that the norm variance  $\text{Var}(\|f(x)\|)$  measures how far the norms of node representations are from their average value. It is different from intra-class variance  $\|\Sigma(f, c)\|_2$  [38], which is the largest eigenvalue of covariance matrix  $\Sigma(f, c)$ . This is also the reason why case 2 is different from case 3 in the previous example.

In order to satisfy the above conditions, we propose a contrastive regularization term, **Contrast-Reg**:

$$\mathcal{L}_{\text{reg}} = -\mathbb{E}_{x, \tilde{x}} \left[ \log \sigma(f(x)^T W r) + \log \sigma(-f(\tilde{x})^T W r) \right], \quad (10)$$

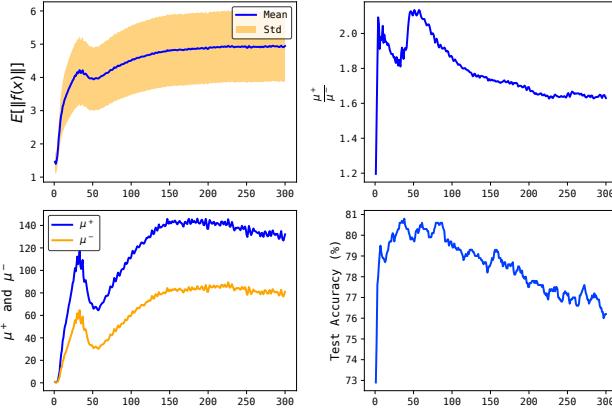
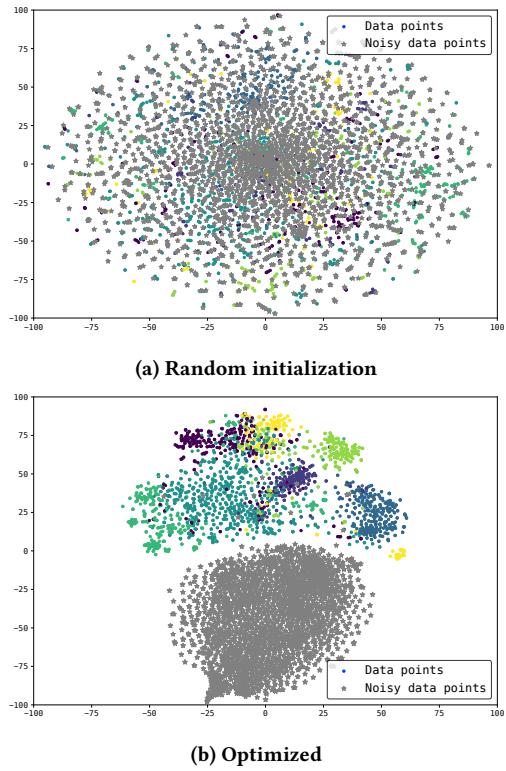


Figure 2: Norm Problem.

Figure 3: t-SNE visualization of  $f(x)^T W$  and  $f(\tilde{x})^T W$ 

where  $\mathbf{r}$  is a random vector uniformly sampled from  $(0, 1]$ ,  $W$  is the trainable parameter, and  $\tilde{x}$  is the noisy features. Different data augmentation techniques such as [9, 46] can be applied to generate the noisy features. In Section 6, we will discuss how we calculate the noisy features in the GNN setting.

We give the motivation of Contrast-Reg's design as follows. Consider an artificial downstream task that is to learn a classifier to discriminate the representations of regular data and noisy data.  $\mathcal{L}_{reg}$  in Eq. (10) can be viewed as the classification loss and  $W$  can

be viewed as the parameter of the bi-linear classifier. The classifier prefers the encoder that can make the representations of intra-class data points more condensed and the inter-class representations more separable. We use the GCN model on the Cora dataset [53] as an example. Figure 3a and Figure 3b show the t-SNE visualization of  $f(x)^T W$  and  $f(\tilde{x})^T W$  before and after the optimization on  $\mathcal{L}_{neg}$ , respectively.

We can observe that the learned representations are closer to each other (i.e., the range of representations in Figure 3a are smaller than that in Figure 3b), while preserving the separability among the representations (i.e., the points with the same label share the same color).

## 5.1 Theoretical Guarantees for Contrast-Reg

Before stating the theorem, we give the following lemma to show that  $\text{Var}(\|f\|)$  can be effectively reduced when  $\|f(x)\|$  is large by adding Contrast-Reg.

**LEMMA 5.1.** *For a random variable  $X \in [1.5, +\infty)$ , a constant  $\tau \in (0, 1]$  and a constant  $c^2$ , we have*

$$\text{Var}\left(\sqrt{\left(X + \frac{\tau}{1+e^X}\right)^2 + c^2}\right) < \text{Var}\left(\sqrt{X^2 + c^2}\right). \quad (11)$$

**PROOF.** First, we consider

$$h(x) = \sqrt{\left(x + \frac{\tau}{1+e^x}\right)^2 + c^2} - \sqrt{x^2 + c^2},$$

where  $h(x)$  is strictly decreasing in  $[x_0, +\infty)$  and strictly increasing in  $(-\infty, x_0]$ , and  $x_0$  is the solution of  $h'(x) = \frac{dh(x)}{dx} = 0$ . Thus, we can approximate the range of  $x_0 \in (0, 1.5)$  by the fact that  $h'(0)h'(1.5) < 0$  for all  $\tau$  and  $c^2$ .

Thus, for  $x > y \geq 1.5$ ,

$$\sqrt{\left(x + \frac{\tau}{1+e^x}\right)^2 + c^2} - \sqrt{x^2 + c^2} < \sqrt{\left(y + \frac{\tau}{1+e^y}\right)^2 + c^2} - \sqrt{y^2 + c^2}$$

and since  $(x + \frac{\tau}{1+e^x})$  is monotonically increasing, we get

$$0 < \sqrt{\left(x + \frac{\tau}{1+e^x}\right)^2 + c^2} - \sqrt{\left(y + \frac{\tau}{1+e^y}\right)^2 + c^2} < \sqrt{x^2 + c^2} - \sqrt{y^2 + c^2}.$$

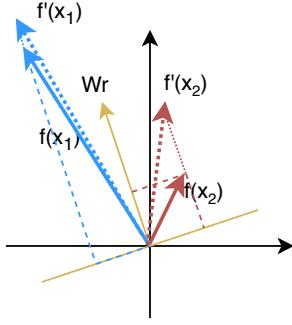
When  $y > x \geq 1.5$ ,

$$\sqrt{x^2 + c^2} - \sqrt{y^2 + c^2} < \sqrt{\left(x + \frac{\tau}{1+e^x}\right)^2 + c^2} - \sqrt{\left(y + \frac{\tau}{1+e^y}\right)^2 + c^2} < 0.$$

Further, we assume that  $X$  and  $Y$  are i.i.d. random variables sampled from  $[1.5, +\infty)$ ,

$$\begin{aligned} & \text{Var}\left(\sqrt{\left(X + \frac{\tau}{1+e^X}\right)^2 + c^2}\right) \\ &= \frac{1}{2} \times \mathbb{E}_{X,Y} \left[ \left( \sqrt{\left(X + \frac{\tau}{1+e^X}\right)^2 + c^2} - \sqrt{\left(Y + \frac{\tau}{1+e^Y}\right)^2 + c^2} \right)^2 \right] \\ &= \frac{1}{2} \times \int \left( \sqrt{\left(x + \frac{\tau}{1+e^x}\right)^2 + c^2} - \sqrt{\left(y + \frac{\tau}{1+e^y}\right)^2 + c^2} \right)^2 p(x)p(y)dxdy \\ &< \frac{1}{2} \times \int \left( \sqrt{x^2 + c^2} - \sqrt{y^2 + c^2} \right)^2 p(x)p(y)dxdy \\ &= \text{Var}(\sqrt{X^2 + c^2}) \end{aligned}$$

□



**Figure 4: Geometric interpretation**

**THEOREM 5.2.** *Minimizing Eq. (10) induces the decrease in  $\text{Var}(\|f(x)\|)$  when  $f(x)^T W \mathbf{r} \in [1.5, +\infty)$ .*

**PROOF.** We minimize  $\mathcal{L}_{\text{neg}}$  by gradient descent with learning rate  $\beta$ .

$$\frac{\partial}{\partial f(x)} \mathcal{L}_{\text{reg}} = -\sigma(-f(x)^T W \mathbf{r}) W \mathbf{r}$$

$$f(x) \leftarrow f(x) + \beta \left( \sigma(-f(x)^T W \mathbf{r}) W \mathbf{r} \right) \quad (12)$$

Eq. (12) shows that in every optimization step,  $f(x)$  extends by  $\beta \sigma(-f(x)^T W \mathbf{r}) \|W \mathbf{r}\|$  along  $\mathbf{r}_0 := \frac{W \mathbf{r}}{\|W \mathbf{r}\|}$ . If we do orthogonal decomposition for  $f(x)$  along  $\mathbf{r}_0$  and its unit orthogonal hyperplane  $\Pi(\mathbf{r}_0)$ ,  $f(x) = (f(x)^T \mathbf{r}_0) \mathbf{r}_0 + (f(x)^T \Pi(\mathbf{r}_0)) \Pi(\mathbf{r}_0)$ . Thus we have

$$\|f(x)\| = \sqrt{(f(x)^T \mathbf{r}_0)^2 + (f(x)^T \Pi(\mathbf{r}_0))^2}. \quad (13)$$

The projection of  $f(x)$  along  $\mathbf{r}_0$  is  $f(x)^T \mathbf{r}_0 = \frac{f(x)^T W \mathbf{r}}{\|W \mathbf{r}\|}$ , while the projection of  $f(x)$  plus the Contrast-Reg update along  $\mathbf{r}_0$  is

$$(f(x)^T \mathbf{r}_0)_{\text{reg}} = \frac{f(x)^T W \mathbf{r}}{\|W \mathbf{r}\|} + \frac{\beta}{1 + e^{f(x)^T W \mathbf{r}}} \|W \mathbf{r}\|.$$

Note that  $(f(x)^T \Pi(\mathbf{r}_0))_{\text{reg}} = f(x)^T \Pi(\mathbf{r}_0)$ .

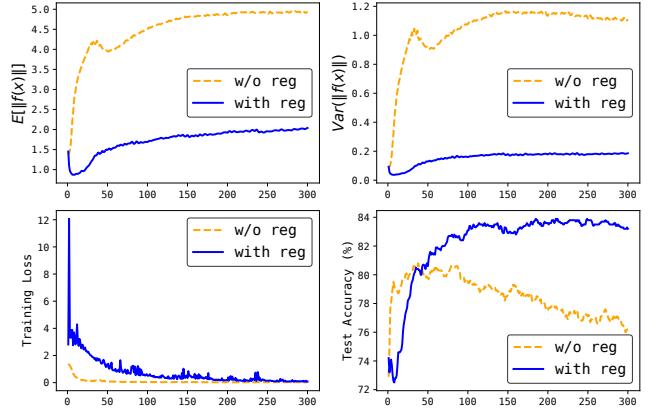
Based on Lemma 5.1 and Eq. (13), when  $\beta \|W \mathbf{r}\|^2 \leq 1$  and  $f(x)^T W \mathbf{r} > 1.5$ , we have

$$\text{Var} \left( \left\| (f(x))_{\text{reg}} \right\| \right) < \text{Var} (\|f(x)\|). \quad (14)$$

□

**REMARK 1.**  $\beta \|W \mathbf{r}\|^2 \leq 1$ , which is the condition of Eq. (14), is not difficult to satisfy, since the magnitude of  $\mathbf{r}$  could be tuned. In practice,  $\mathbf{r} \in (0, 1]$  can fit in all our experiments.

**REMARK 2.** The range of  $f(x)^T W \mathbf{r}$  in Theorem 5.2 is not a tight bound for  $x_0$  in Lemma 5.1. Since when Eq. (10) converges,  $f(x)^T W \mathbf{r}$  is much larger than 1.5 for almost all the samples empirically, we prove the case for  $f(x)^T W \mathbf{r} \in [1.5, +\infty)$ .



**Figure 5: The influence of Contrast-Reg on  $\|f\|$**

## 5.2 Understanding the effects of Contrast-Reg

Theorem 5.2 shows that Contrast-Reg can reduce  $\text{Var}(\|f\|)$  when  $\|f(x)\|$  is large, which is proved from the geometric perspective. Figure 4 visualizes the geometric interpretation of Contrast-Reg. In one gradient descent step,  $f(x)$  and  $f'(x)$  are the representation before and after the gradient descent update of Contrast-Reg. For any data point pair  $x_1$  and  $x_2$ , we decompose  $f(x_1)$  and  $f(x_2)$  along  $W \mathbf{r}$  and its orthogonal direction. Minimizing Eq. (10) is consequently extending  $f(x)$  in each step along  $W \mathbf{r}$  while preserving the length in its orthogonal direction. When we compare  $f(x_1)$  and  $f(x_2)$ , together with  $f'(x_1)$  and  $f'(x_2)$ , we conclude that  $\text{Var}(f'(x)) < \text{Var}(f(x))$ .

Note that the mean and the variance are positively correlated. When we compare  $\mathbb{E}[\|\lambda f\|]$  with  $\mathbb{E}[\|f\|]$ , the former has higher norm and variance for  $|\lambda| > 1$ . Theorem 5.2 shows that our Contrast-Reg can reduce  $\text{Var}(\|f\|)$  when  $\|f(x)\|$  is large, and it should also prefer lower mean because the variance will increase when the norm is scaled to a larger value. Figure 5 shows that the mean and variance are reduced when we apply Contrast-Reg compared to only using one contrastive loss computed by structural similarity. Also, the representation quality is improved significantly.

We also discuss two questions regarding the effects of Contrast-Reg. **Does Contrast-Reg degrade the contrastive learning algorithm to a trivial solution, i.e., all representations converge to one point, even to the origin?** We highlight that Theorem 5.2 is to reduce  $\text{Var}(\|f(x)\|)$  rather than  $\text{Var}(f(x))$ , and Contrast-Reg does not force all the representations into one point. From Figure 4, we know that adding Contrast-Reg only reduces the variance in the representations along the direction of  $W \mathbf{r}$ , while preserving the difference along its orthogonal direction. Therefore, Contrast-Reg not only reduces  $\text{Var}(\|f(x)\|)$ , but also preserves the contrasting ability of the contrastive learning algorithm, as shown in Figure 3b. Furthermore, as we randomize  $\mathbf{r}$  in each training step, the variance reduction on the representation norm is conducted on various directions. Thus, the representations will not have the same dominant direction, and Contrast-Reg does not make the representations converge to one point.

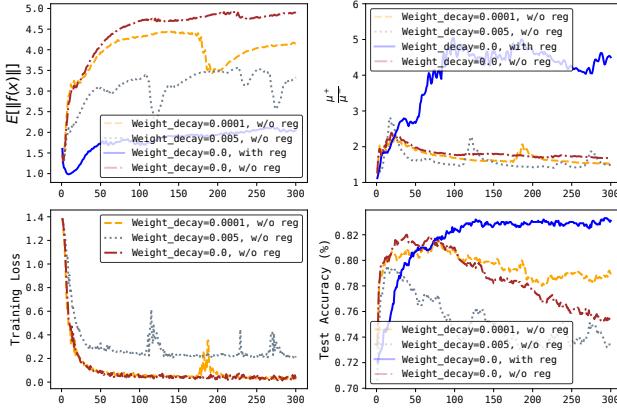


Figure 6: Comparing Contrast-Reg with weight decay

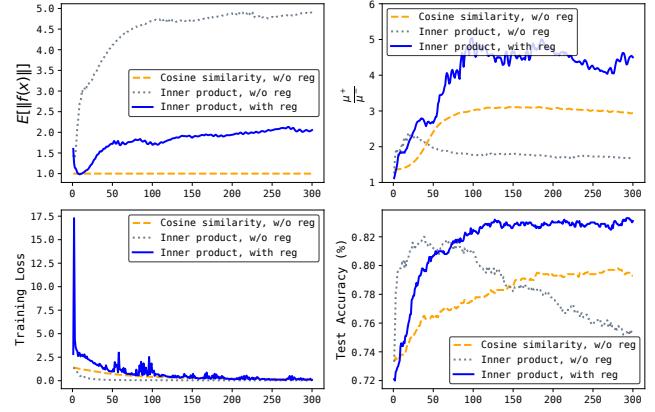
**Can other regularization/normalization methods, e.g., weight decay or final representation normalization, solve the norm problem?** Other regularization and normalization techniques may help stabilize the mean of the representation norms and reduce their variance, but they cannot replace Contrast-Reg as Contrast-Reg leads to more stable changes in the representation norms and preserves high contrast between positive and negative samples.

We compare Contrast-Reg with weight decay and  $\ell_2$ -normalization and show their performance during a 300-epoch training process. Figure 6 shows the performance of weight decay and Contrast-Reg. From the left-top figure, we can see that Contrast-Reg gives smaller and more stable representation norms than weight delay. Specifically, although a large weight decay rate can be applied to obtain smaller representation norms, it leads to fluctuation. The fluctuation in the representation norms impairs the training process, as reflected by the fluctuation in training losses (left-bottom figure). In addition, from the ratio  $\frac{\mu^+}{\mu^-}$ , we know that Contrast-Reg preserves better contrasting ability.

Chen et al. [9] show that adding  $\ell_2$ -normalization (i.e., using cosine similarity rather than inner product) with a temperature parameter improves the representation quality empirically. Figure 7 compares Contrast-Reg with  $\ell_2$ -normalization.  $\ell_2$ -normalization gives much smaller  $\frac{\mu^+}{\mu^-}$  than Contrast-Reg, meaning that the separation between contrastive pairs provided by  $\ell_2$ -normalization is not as clear as that provided by Contrast-Reg. This is because  $\ell_2$ -normalization not only minimizes the variance in the representation norms, but also reduces the differences among the representations, rendering smaller contrast among the representations of data points in different classes. Thus,  $\ell_2$ -normalization gives less improvement in the representation quality than Contrast-Reg.

## 6 A CONTRASTIVE GNN FRAMEWORK

We present our contrastive GNN framework in Algorithm 1. Given a graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  and node attributes  $X$ , we train a GNN model  $f$  for  $e$  epochs. Node representations can be obtained through  $f$  and used as the input to downstream tasks. We adopt NCEloss as the contrastive loss in our framework. For each training epoch, we first select a seed node set  $C$  for computing NCEloss by *SeedSelect*

Figure 7: Comparing Contrast-Reg with  $\ell_2$ -normalization

---

### Algorithm 1: Contrastive GNN Framework

---

```

Input: Graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , node attributes  $X$ , a GNN model
 $f : \mathcal{V} \rightarrow \mathbb{R}^H$ , the number of epochs  $e$ ;
Output: A trained GNN model  $f$ ;
1 Initialize training parameters;
2 for  $epoch \leftarrow 1$  to  $e$  do
3    $C = \text{SeedSelect}(\mathcal{G}, X, f, epoch)$ ;
4    $\mathcal{P} = \text{Contrast}(C, \mathcal{G}, X, f)$ ;
5   loss = NCEloss( $\mathcal{P}$ ) + Contrast-Reg( $\mathcal{G}, X, f$ );
6   Back-propagation and update  $f$ ;
7 end

```

---

(Line 3). Then Line 4 invokes *Contrast* to construct a positive sample and a negative sample for each node in  $C$ , where *Contrast* returns a set  $\mathcal{P}$  of 3-tuples consisting of the representations for the seed nodes, the positive samples and the negative samples, respectively. After that, Line 5 computes the training loss by adding NCEloss on  $\mathcal{P}$  and the regularization loss calculated by Contrast-Reg, and Line 6 updates  $f$  by back-propagation.

As mentioned in Section 5, Contrast-Reg requires noisy features for contrastive regularization. In our contrastive GNN framework, we generate the noisy features by simply shuffling node features among nodes following the corruption function in [46].

For different node similarity definitions, different *SeedSelect* functions are designed to select seed nodes that bring good training effects, while different *Contrast* functions are designed to generate suitable contrastive pairs for the seed nodes. In the following, we demonstrate by examples the designs of three contrastive GNN models for structure, attribute, and proximity similarity, respectively, which are also used in our experimental evaluation in Section 7.

### 6.1 Structure Similarity

We give an example model (*LC*) that captures the community structure inherent in graph data [31]. As clustering is a common and effective method for detecting communities in a graph, we conduct clustering in the node representation space  $f(X)$  to capture community structures. *LC* borrows the design from [24] and implements local clustering by *Contrast* and curriculum learning by

**Algorithm 2: LC**


---

**Hyperparameter:**  $R$ : curriculum update epochs;  $k$ : the number of candidate positive samples for each seed node;

- 1 **Function**  $\text{Contrast}(C, \mathcal{G}, X, f)$ :
- 2   For  $x_i \in C$ , let  $P_i$  be the set of  $k$  nodes in  $\{x_j \in \text{Neighbor}(x_i)\}$  with largest  $f(x_i)^T f(x_j)$ ;
- 3   Randomly pick one positive node  $x_i^+$  from  $P_i$  for each  $x_i \in C$ ;
- 4   Randomly pick one negative node  $x_i^-$  from  $\mathcal{V}$  for each  $x_i \in C$ ;
- 5   **return**  $\{(f(x_i), f(x_i^+), f(x_i^-))\}_{x_i \in C}$ ;
- 6 **end**
- 7 **Function**  $\text{SeedSelect}(\mathcal{G}, X, f, epoch)$ :
- 8   **if**  $epoch \% R \neq 1$  **then**
- 9     **return** the same set of seed nodes  $C$  as in the last epoch ;
- 10   **end**
- 11    $p_{i,j} \leftarrow \frac{f(i)^T f(j)}{\sum_{k \in \mathcal{V}} f(i)^T f(k)}$  for  $i, j \in \mathcal{V}$ ;
- 12    $H(i) \leftarrow -\sum_{j \in \mathcal{V}} (p_{i,j} \log(p_{i,j}))$  for  $i \in \mathcal{V}$ ;
- 13   **return**  $(\lfloor \frac{epoch}{R} \rfloor + 1) \frac{R}{e} |\mathcal{G}|$  nodes with smallest  $H$ ;
- 14 **end**

---

**Algorithm 3: ML**


---

**Parameter:** Parameters of an (additional) GNN layer  $g$ .

- 1 **Function**  $\text{Contrast}(C, \mathcal{G}, X, f)$ :
- 2   Let  $g(x_i)$  be the representation of  $x_i$  by stacking  $g$  upon  $f$ ;
- 3   Randomly pick a negative node  $x_i^-$  from  $\mathcal{V}$  for each  $x_i \in C$ ;
- 4   **return**  $\{(g(x_i), f(x_i), f(x_i^-))\}_{x_i \in C}$ ;
- 5 **end**
- 6 **Function**  $\text{SeedSelect}(\mathcal{G}, X, f, epoch)$ :
- 7   **return**  $\mathcal{V}$ ;
- 8 **end**

---

*SeedSelect*. We remark that other methods such as global clustering [6] and instance discrimination [48] can also be adapted into our contrastive GNN framework by different implementations of *Contrast* and *SeedSelect*.

Algorithm 2 shows the implementation of *Contrast* and *SeedSelect* in LC. For each seed node  $x_i$ , *Contrast* generates a positive node  $x_i^+$  from the nodes that have the highest similarity scores with  $x_i$ , and a negative node  $x_i^-$  randomly sampled from  $V$  (Lines 2-4). *SeedSelect* selects nodes with the smallest entropy to avoid high randomness and uncertainty at the start of the training. For every  $R$  epochs, *SeedSelect* gradually adds more nodes with larger entropy to be computed in the contrastive loss with the progression of epochs (Lines 11-13).

## 6.2 Attribute Similarity

Models adopting attribute similarity assume that nodes with similar attributes are expected to have similar representations, so that the attribute information should be preserved. Hjelm et al. [21], Peng et al. [32] proposed contrastive pair designs to maximize the mutual information between low-level representations (the input features) and high-level representations (the learned representations). Algorithm 3 presents our model, *ML*, which adapts their multi-level representation design into our contrastive GNN framework.

**Table 1: Datasets**

Dataset	Node #	Edge #	Feature #	Class #
Cora	2,708	5,429	1,433	7
Citeseer	3,327	4,732	3,703	6
Pubmed	19,717	44,338	500	3
ogbn-arxiv	169,343	1,166,243	128	40
Wiki	2,405	17,981	4,973	3
Computers	13,381	245,778	767	10
Photo	7,487	119,043	745	8
ogbn-products	2,449,029	61,859,140	100	47
Reddit	232,965	114,615,892	602	41

**Table 2: Contrastive learning with and w/o Contrast-Reg**

Algorithm	Cora	Wiki	Computers	Reddit
ML	73.22 $\pm$ 0.77	58.70 $\pm$ 1.51	77.08 $\pm$ 2.48	94.33 $\pm$ 0.07
ML+reg	<b>82.65</b> $\pm$ 0.57	<b>67.20</b> $\pm$ 0.96	<b>80.30</b> $\pm$ 1.84	<b>94.38</b> $\pm$ 0.04
LC	79.73 $\pm$ 0.75	68.96 $\pm$ 0.56	79.80 $\pm$ 1.49	94.42 $\pm$ 0.03
LC+reg	<b>82.33</b> $\pm$ 0.41	<b>69.19</b> $\pm$ 1.13	<b>80.89</b> $\pm$ 1.64	<b>94.43</b> $\pm$ 0.03
CO	75.49 $\pm$ 0.81	68.52 $\pm$ 1.07	81.02 $\pm$ 1.54	93.85 $\pm$ 0.03
CO+reg	<b>83.63</b> $\pm$ 0.62	<b>70.05</b> $\pm$ 0.98	<b>81.37</b> $\pm$ 1.58	<b>93.92</b> $\pm$ 0.05

In Algorithm 3, *SeedSelect* selects all nodes in a graph as seeds. *Contrast* uses node  $x_i$  itself as the positive node for each seed node  $x_i$ , but in the returned 3-tuple, the representation of  $x_i$  as the seed node is different from the representation of  $x_i$  as the positive node. The second element in the 3-tuple is  $x_i$ 's representation  $f(x_i)$ , while the first element is calculated by stacking an additional GNN layer upon  $f$ . For negative nodes, *Contrast* randomly samples a node in  $V$  for each seed node.

## 6.3 Proximity Similarity

The assumption behind proximity similarity is that nodes are expected to have similar representation when they have high proximity (i.e., they are near neighbors). To capture proximity information among nodes, we implement *SeedSelect* and *Contrast* following the setting in unsupervised GraphSAGE [25]. Adjacent nodes are selected to be positive pairs, while negative pairs are sampled from non-adjacent nodes.

## 7 EXPERIMENTAL RESULTS

In this section, we first show that Contrast-Reg can be generally used for various designs of contrastive pairs. Then we evaluate the performance of various models trained with Contrast-Reg in both graph representation learning and pretraining settings, where contrastive learning is successfully applied.

**Datasets.** The datasets we used include citation networks such as *Cora*, *Citeseer*, *Pubmed* [53] and *ogbn-arxiv* [22], web graphs such as *Wiki* [51], co-purchase networks such as *Computers*, *Photo* [39] and *ogbn-products* [22], and social networks such as *Reddit* [18]. Some statistics of the datasets are given in Table 1.

**Models.** We denote Algorithm 2 (local clustering) capturing *structure similarity* as **ours** (LC), Algorithm 3 (multi-level representations) capturing *attribute similarity* as **ours** (ML) and the algorithm (co-occurrence) capturing *proximity similarity* as **ours** (CO).

**Table 3: Downstream task: node classification**

Algorithm	Cora	Citeseer	Pubmed	ogbn-arxiv	Wiki	Computers	Photo	ogbn-products	Reddit
GCN	81.54±0.68	71.25±0.67	79.26±0.38	<b>71.74±0.29</b>	<b>72.40±0.95</b>	79.82±2.04	<b>88.75±1.99</b>	75.64±0.21	94.02±0.05
node2vec	71.07±0.91	47.37±0.95	66.34±1.40	<b>70.07±0.13</b>	58.76±1.48	75.37±1.52	83.63±1.53	72.49±0.10	93.26±0.04
DGI	81.90±0.84	71.85±0.37	76.89±0.53	69.66±0.18	63.70±1.43	64.92±1.93	77.19±2.60	<b>77.00±0.21</b>	94.14±0.03
GMI	80.95±0.65	71.11±0.15	77.97±1.04	68.36±0.19	63.35±1.03	79.27±1.64	87.08±1.23	75.55±0.39	94.19±0.04
ours (LC)	<b>82.33±0.41</b>	<b>72.88±0.39</b>	<b>79.33±0.59</b>	69.94±0.11	<b>69.19±1.13</b>	<b>80.89±1.64</b>	<b>87.59±1.50</b>	<b>76.96±0.34</b>	<b>94.43±0.03</b>
ours (ML)	<b>82.65±0.57</b>	<b>72.98±0.41</b>	<b>80.10±1.04</b>	<b>70.05±0.09</b>	67.20±0.96	<b>80.30±1.84</b>	86.78±1.70	76.27±0.20	<b>94.38±0.04</b>

**Table 4: Downstream task: graph clustering**

Algorithm	Cora			Citeseer			Wiki		
	Acc	NMI	F1	Acc	NMI	F1	Acc	NMI	F1
node2vec	61.78±0.30	44.47±0.21	62.65±0.26	39.58±0.37	24.23±0.27	37.54±0.39	43.29±0.58	37.39±0.52	36.35±0.51
DGI	<b>71.81±1.01</b>	54.90±0.66	<b>69.88±0.90</b>	68.60±0.47	43.75±0.50	64.64±0.41	44.37±0.92	42.20±0.90	40.16±0.72
AGC	68.93±0.02	53.72±0.04	65.62±0.01	68.37±0.02	42.44±0.03	63.73±0.02	49.54±0.07	47.02±0.09	42.16±0.11
GMI	63.44±3.18	50.33±1.48	62.21±3.46	63.75±1.05	38.14±0.84	60.23±0.79	42.81±0.40	41.53±0.20	38.52±0.22
ours (LC)	70.04±2.04	55.08±0.75	67.36±2.17	67.90±0.74	43.63±0.57	64.21±0.60	50.12±0.96	49.70±0.49	43.74±0.97
ours (ML)	71.59±1.07	<b>56.01±0.64</b>	68.11±1.32	<b>69.17±0.43</b>	<b>44.47±0.46</b>	<b>64.74±0.41</b>	<b>53.13±1.01</b>	<b>51.81±0.57</b>	<b>46.11±0.93</b>

**Unsupervised training procedure.** We used full batch training for Cora, Citeseer, Pubmed, ogbn-arxiv, Wiki, Computers and Photo, while we used stochastic mini-batch training for Reddit and ogbn-products. For Cora, Citeseer, Pubmed, ogbn-arxiv, ogbn-products and Reddit, we used the standard split provided in the datasets and fixed the random seeds from 0 to 9 for 10 different runs. For Computers, Photo and Wiki, we randomly split the train/validation/test as 20 nodes/30 nodes/all the remaining nodes per class, as recommended in [39]. The performance was measured on 25 ( $5 \times 5$ ) different runs, with 5 random splits and 5 fixed-seed runs (from 0 to 4) for each random split. For Wiki, we removed the edge attributes for all models for fair comparison. The additional special designs for link prediction task and pretraining setting are given in their respective subsections.

## 7.1 Generalizability of Contrast-Reg

To evaluate the performance gain by Contrast-Reg, we tested the model performance (on node classification accuracy) with and without Contrast-Reg on four networks from four different domains. GCN encoder was used on Cora, Wiki and Computers. Graph-Sage with GCN-aggregation encoder was used on Reddit. Table 2 shows that Contrast-Reg can help better capture different types of similarity, i.e., **ML** for attribute similarity, **LC** for structure similarity, and **CO** for proximity similarity, and improve the performance of the models in all cases. In the following experiments, we omit CO since its contrast loss is computed by sampled edges and thus the computation cost is larger than the other two contrast designs.

## 7.2 Graph Representation Learning

Next we show that high quality representations can be learned by our method. High quality means using these representations, simple models (e.g., linear classifier for classification and  $k$ -means for clustering) can easily achieve high performance on various downstream tasks. To show that, we evaluated the learned representations on

three downstream tasks that are fundamental for graph analytics: node classification, graph clustering, and link prediction.

**7.2.1 Node Classification.** We evaluated the performance of node classification on all datasets, using both full batch training and stochastic mini-batch training. We compared our methods with DGI [46], GMI [32], node2vec [16], and supervised GCN [25]. DGI and GMI are the state-of-the-art algorithms in unsupervised graph representation learning. Node2vec is the representative algorithm for random walk based graph representation algorithms [16, 33, 41]. GCN is a classic supervised GNN model. We report ours (LC) and ours (ML), both using Contrast-Reg, where their GNN encoder is GCN for full batch training and GraphSage [18] with GCN-aggregation for stochastic training, respectively. The encoder settings are the same as in DGI and GMI. Our framework can also adopt other encoder such as GAT and similar performance improvements over GAT can also be obtained. We omit the detailed results due to the page limit.

Table 3 reports node classification accuracy with standard deviation. The results show that our algorithms achieve better performance in the majority of the cases, for both full batch training (on Cora, Citeseer, Pubmed, Computers, Photo and Wiki) and stochastic training (on Reddit and Ogbn-products). Our unsupervised algorithms can even outperform the supervised GCN. Compared with DGI and GMI, our model is similar to the DGI model with a properly designed contrast pair and to the GMI model with the Contrast-Reg term, and thus we can achieve better performance in most cases. If we compare Table 2 and Table 3, it shows that the performance gain is from Contrast-Reg rather than a more proper contrast design.

**7.2.2 Graph Clustering.** Following the work of Xia et al. [49], we used three metrics to evaluate the clustering performance: accuracy (Acc), normalized mutual information (NMI), and F1-macro (F1). For all these three metrics, a higher value indicates better clustering performance. We compared our methods with DGI, node2vec, GMI,

**Table 5: Downstream task: link prediction**

Algorithm	Cora	Citeseer	Pubmed	Wiki
GCN-neg	92.40 $\pm$ 0.51	92.27 $\pm$ 0.90	97.24 $\pm$ 0.19	93.27 $\pm$ 0.31
node2vec	86.33 $\pm$ 0.87	79.60 $\pm$ 1.58	81.74 $\pm$ 0.57	92.41 $\pm$ 0.35
DGI	93.62 $\pm$ 0.98	95.03 $\pm$ 1.73	97.24 $\pm$ 0.13	95.55 $\pm$ 0.35
GMI	91.31 $\pm$ 0.88	92.23 $\pm$ 0.80	95.14 $\pm$ 0.25	95.30 $\pm$ 0.29
ours (LC)	<b>94.61<math>\pm</math>0.64</b>	<b>95.63<math>\pm</math>0.88</b>	<b>97.26<math>\pm</math>0.15</b>	<b>96.28<math>\pm</math>0.21</b>

**Table 6: Pretraining**

Algorithm	Reddit	ogbn-products
No pretraining	90.44 $\pm$ 1.62	84.69 $\pm$ 0.79
DGI	92.09 $\pm$ 1.05	86.37 $\pm$ 0.19
GMI	92.13 $\pm$ 1.16	86.14 $\pm$ 0.16
ours (ML)	92.18 $\pm$ 0.97	86.28 $\pm$ 0.20
ours (LC)	<b>92.52<math>\pm</math>0.55</b>	<b>86.45<math>\pm</math>0.13</b>

and AGC [55] on Cora, Citeseer and Wiki. AGC [55] is a state-of-the-art graph clustering algorithm, which exploits high-order graph convolution to do attribute graph clustering. For all models and all datasets, we used  $k$ -means to cluster both the labels and representations of nodes. The clustering results of labels are taken as the ground truth. Since high dimension is harmful to clustering [8], we applied the PCA algorithm to the representations to reduce the dimensionality before using  $k$ -means. The random seed setting for model training was the same as that in the node classification task. And to reduce the randomness caused by  $k$ -means, we set the random seed of clustering from 0 to 4, and took the average result for each learned representations. For each cell in Table 4, we report the better result with PCA and without PCA. The results show that our algorithms, especially ours (ML), achieve better performance in most cases, which again demonstrates the effectiveness of Contrast-Reg. Note that graph clustering is applied on attribute graphs, the fact that the results of ours (ML) are better than ours (LC) tells us that attributes play an important role in clustering.

**7.2.3 Link Prediction.** The representations learned in Sections 7.2.1 and 7.2.2 should not be directly used in the link prediction task, because the encoder already has access to all edges in an input graph when we train it using contrastive learning, which leads to the data linkage issue (i.e., the edges used in the prediction task being accessible in the training process). Thus, for link prediction, an inductive setting of graph representation learning was adopted. We extracted random induced subgraphs (85% of the edges) from each origin graph to train the representation learning model and the link predictor. The remaining edges were used to validate and test the link prediction results (10% of the edges as the test edge set, 5% as the validation edge set). The performance was evaluated on 25 (5x5) different runs, with 5 different induced subgraphs (fixed-seed random split scheme) and 5 fixed-seed runs (from 0 to 4). we compared our model with DGI, GMI, node2vec and unsupervised GCN (i.e., GCN-neg in Table 5) on Cora, Citeseer, Pubmed and Wiki. The results in Table 5 show that our algorithms achieve better performance than the state-of-the-art methods. We did not conduct the ML model in this experiment because ML pays more attention to the node attributes.

### 7.3 Pretraining

We also evaluated the performance of Contrast-Reg for pretraining. For the Reddit dataset, as it can be naturally split by time, we pretrained the models using the first 20 days (by generating an induced subgraph based on the pretraining nodes) from the dataset, the remaining data was split into three parts: the first part generated a new subgraph for fine-tuning the pretrained model and training the classifier, and the second and third parts were used for validation and test. For the ogbn-products dataset, we split the dataset based on node id, i.e., pretraining the models using a subgraph generated by the first 70% nodes, where the data splitting scheme for the remaining data is the same as the Reddit dataset. The baseline experiments were conducted on DGI and GMI with the same GraphSage with GCN-aggregation encoder as in our model. Table 6 shows that pretraining the model helps the model converge to a better representation model with low variance, and Contrast-Reg can improve the transferability of pretraining a model.

## 8 CONCLUSIONS

We showed that the high scales of node representations’ norms and high variance among them could make contrastive learning algorithms fail. We then proposed Contrast-Reg to avoid the cases that are harmful to the representation quality and showed from the geometric perspective that Contrast-Reg stabilizes the scales of norms and reduces their variance. Our experiments validated that Contrast-Reg improves the representation quality.

## REFERENCES

- [1] Amr Ahmed, Nino Shervashidze, Shravan M. Narayananamurthy, Vanja Josifovski, and Alexander J. Smola. 2013. Distributed large-scale natural graph factorization. In *WWW ’13*. 37–48.
- [2] Yuki M. Asano, Christian Rupprecht, and Andrea Vedaldi. 2020. Self-labelling via simultaneous clustering and representation learning. In *ICLR ’20*.
- [3] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, R. Devon Hjelm, and Aaron C. Courville. 2018. Mutual Information Neural Estimation. In *ICML ’18*. 530–539.
- [4] Stephen P. Boyd and Lieven Vandenberghe. 2014. *Convex Optimization*.
- [5] Shaosheng Cao, Wei Lu, and Qiongkai Xu. 2015. GraRep: Learning Graph Representations with Global Structural Information. In *CIKM ’15*. 891–900.
- [6] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. 2018. Deep Clustering for Unsupervised Learning of Visual Features. In *ECCV ’18*. 139–156.
- [7] Haochen Chen, Bryan Perozzi, Yifan Hu, and Steven Skiena. 2018. HARP: Hierarchical Representation Learning for Networks. In *(AAAI) ’18, (IAMI) ’18, and (EAAI) ’18*. 2127–2134.
- [8] Lei Chen. 2009. Curse of Dimensionality. In *Encyclopedia of Database Systems*. 545–546.
- [9] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A Simple Framework for Contrastive Learning of Visual Representations. *CoRR* abs/2002.05709 (2020). arXiv:2002.05709
- [10] Zhijie Deng, Yinpeng Dong, and Jun Zhu. 2019. Batch Virtual Adversarial Training for Graph Convolutional Networks. *CoRR* abs/1902.09192 (2019). arXiv:1902.09192
- [11] Ming Ding, Jie Tang, and Jie Zhang. 2018. Semi-supervised Learning on Graphs with Generative Adversarial Nets. In *CIKM ’18*. 913–922.
- [12] Claire Donnat, Marinka Zitnik, David Hallac, and Jure Leskovec. 2018. Learning Structural Node Embeddings via Diffusion Wavelets. In *KDD ’18*. 1320–1329.
- [13] Chris Dyer. 2014. Notes on Noise Contrastive Estimation and Negative Sampling. (2014). arXiv:1410.8251
- [14] Fuli Feng, Xiangnan He, Jie Tang, and Tat-Seng Chua. 2019. Graph Adversarial Training: Dynamically Regularizing Based on Graph Structure. *CoRR* abs/1902.08226 (2019). arXiv:1902.08226
- [15] Matthias Fey and Jan Eric Lenssen. 2019. Fast Graph Representation Learning with PyTorch Geometric. *CoRR* abs/1903.02428 (2019). arXiv:1903.02428
- [16] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable Feature Learning for Networks. In *KDD ’16*. 855–864.

- [17] Michael Gutmann and Aapo Hyvärinen. 2012. Noise-Contrastive Estimation of Unnormalized Statistical Models, with Applications to Natural Image Statistics. *J. Mach. Learn. Res.* 13 (2012), 307–361.
- [18] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *NeurIPS ’17*. 1024–1034.
- [19] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. 2020. Momentum Contrast for Unsupervised Visual Representation Learning. In *CVPR ’20*. 9726–9735.
- [20] Olivier J. Hénaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, S. M. Ali Eslami, and Aäron van den Oord. 2019. Data-Efficient Image Recognition with Contrastive Predictive Coding. *CoRR* abs/1905.09272 (2019). arXiv:1905.09272
- [21] R. Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Philip Bachman, Adam Trischler, and Yoshua Bengio. 2019. Learning deep representations by mutual information estimation and maximization. In *ICLR ’19*.
- [22] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. 2020. Open Graph Benchmark: Datasets for Machine Learning on Graphs. *CoRR* abs/2005.00687 (2020). arXiv:2005.00687
- [23] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay S. Pande, and Jure Leskovec. 2020. Strategies for Pre-training Graph Neural Networks. In *ICLR ’20*.
- [24] Jiabo Huang, Qi Dong, Shaogang Gong, and Xiatian Zhu. 2019. Unsupervised Deep Learning by Neighbourhood Discovery. In *ICML ’19*. 2849–2858.
- [25] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR ’17*.
- [26] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *ICLR ’19*.
- [27] Andreas Maurer. 2016. A Vector-Contraction Inequality for Rademacher Complexities. In *ALT ’16*, Ronald Ortner, Hans Ulrich Simon, and Sandra Zilles (Eds.). 3–17.
- [28] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *NeurIPS ’13*. 3111–3119.
- [29] Andriy Mnih and Yee Whye Teh. 2012. A fast and simple algorithm for training neural probabilistic language models. In *ICML ’12*.
- [30] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. 2018. *Foundations of Machine Learning* (2nd ed.). The MIT Press.
- [31] M. E. J. Newman. 2006. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences* 103, 23 (2006), 8577–8582.
- [32] Zhen Peng, Wenbing Huang, Minnan Luo, Qinghua Zheng, Yu Rong, Tingyang Xu, and Junzhou Huang. 2020. Graph Representation Learning via Graphical Mutual Information Maximization. In *WWW ’20*. 259–270.
- [33] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. DeepWalk: online learning of social representations. In *KDD ’14*. 701–710.
- [34] Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. 2020. GCC: Graph Contrastive Coding for Graph Neural Network Pre-Training. In *KDD ’20*. 1150–1160.
- [35] Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. 2018. Network Embedding as Matrix Factorization: Unifying DeepWalk, LINE, PTE, and node2vec. In *WSDM ’18*. ACM, 459–467.
- [36] Meng Qu, Yoshua Bengio, and Jian Tang. 2019. GMNN: Graph Markov Neural Networks. In *ICML ’19*, Vol. 97. 5241–5250.
- [37] Leonardo Filipe Rodrigues Ribeiro, Pedro H. P. Saverese, and Daniel R. Figueiredo. 2017. struc2vec: Learning Node Representations from Structural Identity. In *KDD ’17*. 385–394.
- [38] Nikunj Saunshi, Orestis Plevrakis, Sanjeev Arora, Mikhail Khodak, and Hrishikesh Khandeparkar. 2019. A Theoretical Analysis of Contrastive Unsupervised Representation Learning. In *ICML ’19*. 5628–5637.
- [39] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. 2018. Pitfalls of Graph Neural Network Evaluation. *CoRR* abs/1811.05868 (2018). arXiv:1811.05868
- [40] Fan-Yun Sun, Meng Qu, Jordan Hoffmann, Chin-Wei Huang, and Jian Tang. 2019. vGraph: A Generative Model for Joint Community Detection and Node Representation Learning. In *NeurIPS ’19*. 512–522.
- [41] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. LINE: Large-scale Information Network Embedding. In *WWW ’15*. 1067–1077.
- [42] Yonglong Tian, Dilip Krishnan, and Phillip Isola. 2019. Contrastive Multiview Coding. *CoRR* abs/1906.05849 (2019). arXiv:1906.05849
- [43] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. 2020. What makes for good views for contrastive learning. *CoRR* abs/2005.10243 (2020). arXiv:2005.10243
- [44] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation Learning with Contrastive Predictive Coding. *CoRR* abs/1807.03748 (2018). arXiv:1807.03748
- [45] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *ICLR ’18*.
- [46] Petar Velickovic, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R. Devon Hjelm. 2019. Deep Graph Infomax. In *ICLR ’19*.
- [47] Minjie Wang, Da Zheng, Zihao Ye, Quan Gan, Mufei Li, Xiang Song, Jinjing Zhou, Chao Ma, Lingfan Yu, Yu Gai, Tianjun Xiao, Tong He, George Karypis, Jinyang Li, and Zheng Zhang. 2019. Deep Graph Library: A Graph-Centric, Highly-Performant Package for Graph Neural Networks. *arXiv preprint arXiv:1909.01315* (2019).
- [48] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. 2018. Unsupervised Feature Learning via Non-Parametric Instance-level Discrimination. *CoRR* abs/1805.01978 (2018). arXiv:1805.01978
- [49] Rongkai Xia, Yan Pan, Lei Du, and Jian Yin. 2014. Robust Multi-View Spectral Clustering via Low-Rank and Sparse Decomposition. In *AAAI ’14*. 2149–2155.
- [50] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How Powerful are Graph Neural Networks?. In *ICLR ’19*.
- [51] Cheng Yang, Zhiyuan Liu, Deli Zhao, MaoSong Sun, and Edward Y. Chang. 2015. Network Representation Learning with Rich Text Information. In *IJCAI ’15*. 2111–2117.
- [52] Han Yang, Kaili Ma, and James Cheng. 2020. Rethinking Graph Regularization For Graph Neural Networks. *arXiv:2009.02027*
- [53] Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. 2016. Revisiting Semi-Supervised Learning with Graph Embeddings. In *ICML ’16 (JMLR Workshop and Conference Proceedings, Vol. 48)*. 40–48.
- [54] Zhen Yang, Ming Ding, Chang Zhou, Hongxia Yang, Jingren Zhou, and Jie Tang. 2020. Understanding Negative Sampling in Graph Representation Learning. In *KDD ’20*. 1666–1676.
- [55] Xiaotong Zhang, Han Liu, Qimai Li, and Xiao-Ming Wu. 2019. Attributed Graph Clustering via Adaptive Graph Convolution. In *IJCAI ’19*. 4327–4333.

## A PROOFS

### A.1 Proof of Lemma 4.2

First, we prove that  $\ell(f(x^+), \{f(x_i^-)\}) = -(\log \sigma(f(x)^T f(x^+))) + \sum_{i=1}^K \log(\sigma(f(x)^T f(x_i^-)))$  is convex w.r.t.  $f(x^+), f(x_1^-), \dots, f(x_K^-)$ . Consider that  $\ell_1(z) = -\log \sigma(z)$  and  $\ell_2(z) = -\log \sigma(-z)$  are both convex functions since  $\ell_1'' > 0$  and  $\ell_2'' > 0$  for  $z \in \mathbb{R}$ . Given  $f(x) \in \mathbb{R}$ ,  $z^+ = f(x)^T f(x^+)$  and  $z^- = f(x)^T f(x^+)$  are affine transformation w.r.t.  $f(x^+)$  and  $f(x^-)$ . Thus, when  $f(x)$  is fixed,  $\ell_1(f(x^+)) = -\log \sigma(f(x)^T f(x^+))$  and  $\ell_2(f(x^-)) = -\log \sigma(-f(x)^T f(x^-))$  are convex functions. As  $\ell_1 > 0$  and  $\ell_2 > 0$ , we obtain  $\ell(f(x^+), \{f(x_i^-)\}) = -(\log \sigma(f(x)^T f(x^+)) + \sum_{i=1}^K \log(\sigma(f(x)^T f(x_i^-))))$  is convex since non-negative weighted sums preserve convexity [4]. By the definition of convexity,

$$\begin{aligned} \mathcal{L}_{nce}(f) &= \mathbb{E}_{c^+, c^- \sim \rho^2; \substack{x \in \mathcal{D}_{c^+} \\ x \in \mathcal{D}_{c^-}}} [\ell(f(x^+), \{f(x_i^-)\})] \\ &\geq \mathbb{E}_{c^+, c^- \sim \rho^2; x \sim \mathcal{D}_{c^+}} [\ell(f(x)^T (\mu_{c^+} - \mu_{c^-}))] \\ &= (1 - \tau) \mathcal{L}_{sup}^\mu(f) + \tau \end{aligned}$$

### A.2 Generalization bound

Denote

$$\tilde{\mathcal{F}} = \left\{ \tilde{f}(x_i, x_i^+, x_{i1}^-, \dots, x_{iK}^-) = (f(x_i), f(x_i^+), f(x_{i1}^-), \dots, f(x_{iK}^-)) \mid f \in \mathcal{F} \right\}.$$

Let  $q_{\tilde{f}} = h \circ \tilde{f}$ , and its function class,

$$\mathcal{Q} = \left\{ q = h \circ \tilde{f} \mid \tilde{f} \in \tilde{\mathcal{F}} \right\}.$$

Denote  $z_i = (x_i, x_i^+, x_{i1}^-, \dots, x_{iK}^-)$ , suppose  $\ell$  is bounded by  $B$ , then we can decompose  $h = \frac{1}{B} \ell \circ \phi$ . Then we have  $q_{\tilde{f}}(z_i) = \frac{1}{B} \ell(\phi(\tilde{f}(z_i)))$ ,

where

$$\begin{aligned}\phi(\tilde{f}(z_i)) &= \left( \sum_{t=1}^d f(x_i)_t f(x_{i0}^+_t), \sum_{t=1}^d f(x_i)_t f(x_{i1}^-_t), \dots, \right. \\ &\quad \left. \sum_{t=1}^d f(x_i)_t f(x_{ik}^-_t) \right) \\ \ell(\mathbf{x}) &= -\left( \log \sigma(x_0) + \sum_{i=1}^K \log \sigma(-x_i) \right).\end{aligned}\quad (15)$$

From Eq. (15), we know that  $\phi : \mathbb{R}^{(K+2)d} \rightarrow \mathbb{R}^{K+1}$ .

Then we will prove that  $h$  is  $L$ -Lipschitz by proving that  $\phi$  and  $\ell$  are both Lipschitz continuity. First,

$$\begin{aligned}\frac{\partial \phi(\tilde{f}(z_i))}{\partial f(x_i)_t} &= f(x_{ik})_t = \begin{cases} f(x_{i0}^+_t), & k=0 \\ f(x_{ik}^-_t), & k=1, \dots, K \end{cases} \\ \frac{\partial \phi(\tilde{f}(z_i))}{\partial f(x_{i0}^+_t)} &= f(x_i)_t, \quad \frac{\partial \phi(\tilde{f}(z_i))}{\partial f(x_{ik}^-_t)} = f(x_i)_t.\end{aligned}$$

If we assume  $\sum_{t=1}^d f(x_{ik})_t^2 \leq R^2$  and  $\sum_{t=1}^d f(x_i)_t^2 \leq R^2$ ,

$$\begin{aligned}\|J\|_F &= \sqrt{\sum_{t=1}^d f(x_{i0}^+_t)^2 + \sum_{k=1}^K \sum_{t=1}^d f(x_{ik}^-_t)^2 + (K+1) \sum_{t=1}^d f(x_i)_t^2} \\ &\leq \sqrt{2(K+1)R^2} = \sqrt{2(K+1)}R\end{aligned}$$

Combining  $\|J\|_2 \leq \|J\|_F$ , we obtain that  $\phi$  is  $\sqrt{2(K+1)}R$ -Lipschitz. Similarly,  $\ell$  is  $\sqrt{K+1}$ -Lipschitz. Since we assume that the inner product of embedding is no more than  $R^2$ . Thus,  $l$  is bounded by  $B = -(K+1)\log(\sigma(-R^2))$ . Above all,  $h$  is  $L$ -Lipschitz with  $L = \frac{\sqrt{2(K+1)}R}{B}$ . Applying vector-contraction inequality[27]. We have

$$\mathbb{E}_{\sigma \sim \{\pm 1\}^M} [\sup_{\tilde{f} \in \tilde{\mathcal{F}}} \langle \sigma, (h \circ \tilde{f})|_S \rangle] \leq \sqrt{2}L \mathbb{E}_{\sigma \sim \{\pm 1\}^{(K+1)dM}} [\sup_{\tilde{f} \in \tilde{\mathcal{F}}} \langle \sigma, \tilde{f}|_S \rangle].$$

If we write it in Rademacher complexity manner, we have

$$\mathcal{R}_S(Q) \leq \frac{2(K+1)R}{B} \mathcal{R}_S(\mathcal{F}).$$

Applying generalization bounds based on Rademacher complexity [30] to  $q \in Q$ . For any  $\delta > 0$ , with the probability of at least  $1 - \frac{\delta}{2}$ ,

$$\begin{aligned}\mathbb{E}[q(\mathbf{z})] &\leq \frac{1}{M} \sum_{i=1}^M q(\mathbf{z}_i) + \frac{2\mathcal{R}_S(Q)}{M} + 3\sqrt{\frac{\log \frac{4}{\delta}}{2M}} \\ &\leq \frac{1}{M} \sum_{i=1}^M q(\mathbf{z}_i) + \frac{4(K+1)R\mathcal{R}_S(\mathcal{F})}{BM} + 3\sqrt{\frac{\log \frac{4}{\delta}}{2M}}.\end{aligned}$$

Thus for any  $f$ ,

$$\mathcal{L}_{nce}(f) \leq \tilde{\mathcal{L}}_{nce}(f) + \frac{4(K+1)R\mathcal{R}_S(\mathcal{F})}{M} + 3B\sqrt{\frac{\log \frac{4}{\delta}}{2M}}. \quad (16)$$

Let  $\hat{f} = \arg \min_{f \in \mathcal{F}} \tilde{\mathcal{L}}_{nce}(f)$  and  $f^* = \arg \min_{f \in \mathcal{F}} \mathcal{L}_{nce}(f)$ . By Hoeffding's inequality, with probability of  $1 - \frac{\delta}{2}$ ,

$$\tilde{\mathcal{L}}_{nce}(f^*) \leq \mathcal{L}_{nce}(f^*) + B\sqrt{\frac{\log \frac{2}{\delta}}{2M}} \quad (17)$$

Substituting  $\hat{f}$  into Eq. (16), combining  $\tilde{\mathcal{L}}_{nce}(\hat{f}) \leq \mathcal{L}_{nce}(f^*)$  and applying union bound, with probability of at most  $\delta$

$$\begin{aligned}\mathcal{L}_{nce}(\hat{f}) &\leq \tilde{\mathcal{L}}_{nce}(\hat{f}) + \frac{4(K+1)R\mathcal{R}_S(\mathcal{F})}{M} + 3B\sqrt{\frac{\log \frac{4}{\delta}}{2M}} + B\sqrt{\frac{\log \frac{2}{\delta}}{2M}} \\ &\leq \mathcal{L}_{nce}(f^*) + \frac{4(K+1)R\mathcal{R}_S(\mathcal{F})}{M} + 4B\sqrt{\frac{\log \frac{4}{\delta}}{2M}} \\ &\leq \mathcal{L}_{nce}(f) + \frac{4(K+1)R\mathcal{R}_S(\mathcal{F})}{M} - 4(K+1)\log(\sigma(-R^2))\sqrt{\frac{\log \frac{4}{\delta}}{2M}}\end{aligned}\quad (18)$$

fails. Thus, with probability of at least  $1 - \delta$ , Eq. (18) holds.

### A.3 Class collision loss

Let  $p_i = |f(x)^T f(x_i)|$  and  $p = \max_{i \in \{0, 1, \dots, K\}} p_i$ . Considering

$$\begin{aligned}\mathcal{L}_{nce}^-(f) &= -\mathbb{E} \left[ \log \sigma(f(x)^T f(x_0^+)) + \sum_{i=1}^K \log \sigma(-f(x)^T f(x_i^-)) \right] \\ &= \mathbb{E} \left[ \log(1 + e^{-f(x)^T f(x_0^+)}) + \sum_{i=1}^K \log(1 + e^{f(x)^T f(x_i^-)}) \right] \\ &\leq (K+1)\mathbb{E}[\log(1 + e^p)] \\ &\leq (K+1)\log c' + (K+1)\mathbb{E}[p]\end{aligned}\quad (19)$$

where  $c' \in [1 + e^{-R^2}, 2]$ .

Since  $x, x_0^+, x_1^-, \dots, x_K^-$  are sampled i.i.d. from the same class,

$$\mathbb{E}[p] = \int P[p \geq x] dx = \int (1 - (1 - P[p_0 \geq x])^{K+1}) dx. \quad (20)$$

Applying Bernoulli's inequality, we have

$$\begin{aligned}\mathbb{E}[p] &\leq \int (1 - (1 - (K+1)P[p_0 \geq x])) dx \\ &= \int (K+1)P[p_0 \geq x] dx \\ &= (K+1)\mathbb{E}[p_0] \\ &= (K+1)\mathbb{E}[|f(x)^T f(x_0^+)|] \\ &\leq (K+1)\sqrt{\mathbb{E}[(f(x)^T f(x_0^+))^2]}.\end{aligned}\quad (21)$$

Therefore,

$$\mathcal{L}_{nce}^-(f) \leq (K+1)\log c' + (K+1)^2 s(f) \quad (22)$$

## B EXPERIMENT DETAILS

**Hardware Configuration:** The experiments are conducted on Linux servers installed with an Intel(R) Xeon(R) Silver 4114 CPU @ 2.20GHz, 256GB RAM and 8 NVIDIA 2080Ti GPUs.

**Software Configuration:** Our models, as well as the DGI, GMI and GCN baselines, were implemented in PyTorch Geometric [15] version 1.4.3, DGL [47] version 0.5.1 with CUDA version 10.2, scikit-learn version 0.23.1 and Python 3.6. Our codes and datasets will be made available.

**Hyper-parameters:** For full batch training, we used 1-layer GCN as the encoder with prelu activation, for mini-batch training, we used a 3-layer GCN with prelu activation. We conducted grid search of different learning rate (from 1e-2, 5e-3, 3e-3, 1e-3, 5e-4, 3e-4, 1e-4) and curriculum settings (including learning rate decay and curriculum rounds) on the fullbatch version. We used 1e-3 or 5e-4 as the learning rate; 10,10,15 or 10,10,25 as the fanouts and 1024 or 512 as the batch size for mini-batch training.