Calibrating and Improving Graph Contrastive Learning

Anonymous authors
Paper under double-blind review

Abstract

Graph contrastive learning algorithms have demonstrated remarkable success in various applications such as node classification, link prediction, and graph clustering. However, directly applying contrastive pairs and graph neural network (GNN) models to new tasks or data has proven to be inconsistently effective. In this paper, we first adapt the expected calibration error (ECE) to the graph contrastive learning framework to assess the quality of embeddings and their accuracy in downstream tasks. We identify miscalibration issues in existing algorithms and propose a novel regularization method, **Contrast-Reg**, to address these limitations. By ensuring node representations maintain similarity with a random vector and that pseudo-negative representations, generated through shuffled node features, remain dissimilar to the same random vector, Contrast-Reg guarantees that minimizing the contrastive loss results in high-quality representations that improve accuracy in downstream tasks, rather than overfitting specific spurious features. We provide both theoretical evidence and empirical experiments to support the effectiveness of Contrast-Reg, demonstrating its ability to enhance generalizability and improve the performance of graph contrastive algorithms with different similarity definitions across various downstream tasks.

1 Introduction

Graph structures are widely used to capture abundant information, such as hierarchical configurations and community structures, present in data from various domains like social networks, e-commerce networks, knowledge graphs, the World Wide Web, and semantic webs. By incorporating graph topology along with node and edge attributes into machine learning frameworks, graph representation learning has demonstrated remarkable success in numerous essential applications, such as node classification, link prediction, and graph clustering. A large number of graph representation learning algorithms Velickovic et al. (2019); Peng et al. (2020); Tang et al. (2015); Perozzi et al. (2014); Grover & Leskovec (2016); Ahmed et al. (2013); Cao et al. (2015); Qiu et al. (2018); Chen et al. (2018); Kipf & Welling (2017); Hamilton et al. (2017); Velickovic et al. (2018); Xu et al. (2019); Qu et al. (2019) have been proposed. Among them, many Tang et al. (2015); Perozzi et al. (2014); Grover & Leskovec (2016); Hamilton et al. (2017) are designed in an unsupervised manner and utilize *negative sampling*" to learn node representations. This design shares similar ideas with *contrastive learning* He et al. (2020); Tian et al. (2020; 2019); van den Oord et al. (2018); Hénaff et al. (2019); Belghazi et al. (2018); Hjelm et al. (2019); Wu et al. (2018); Mikolov et al. (2013); Asano et al. (2020); Caron et al. (2018); Chen et al. (2020), which contrasts" the similarities of the representations of similar (or positive) node pairs against those of negative pairs. These algorithms employ *noise contrastive estimation loss* (*NCEloss*), differing in their definition of node similarity (hence the design of contrastive pairs) and the design of the encoder backbone. After getting the embeddings outputted by the graph contrastive algorithms, these embeddings could be directly delivered to the downstream tasks. Although graph contrastive algorithms have demonstrated strong performance in some downstream tasks, we have discovered that directly applying contrastive pairs and GNN models, such as graph convolutional networks (GCNs) Kipf & Welling (2017), to new tasks or data is not consistently effective (as shown in Section 6). To address this issue, we first adapt the expected calibration error (ECE) to the graph contrastive learning framework to assess the quality of the embeddings generated by the model, $\sigma(h_v \cdot h_{v'})$, and their accuracy in downstream tasks, $acc(v, v')$. We observe that the model learns certain spurious features that reduce the loss but ultimately prove detrimental to downstream task performance when solely minimizing the NCE loss. Furthermore, we identify two factors that contribute to the model's miscalibration: a) the expectation of the prediction value for randomly sampled pairs, $\mathbb{E}(v, v')[\sigma(h_v \cdot h_{v'})]$, and b) the probability q^+ of v'_+ sharing the same label as v in positive sampling. To address the miscalibration in existing graph contrastive learning algorithms, we introduce a novel regularization method, denoted as **Contrast-Reg**. Contrast-Reg employs a regularization vector r, which consists of a random vector with each entry within the range $(0, 1]$. By ensuring node representations maintain similarity with r and that pseudo-negative representations, generated through shuffled node features, remain dissimilar to r, Contrast-Reg guarantees that minimizing the contrastive loss results in high-quality representations that improve accuracy in downstream tasks, rather than overfitting specific spurious features. We provide both theoretical evidence and empirical experiments to support the effectiveness of Contrast-Reg. First, we derive a generalization bound for our contrastive GNN framework, building upon the theoretical framework presented in Saunshi et al. (2019), and demonstrate that Contrast-Reg contributes to a decrease in the upper bound. This result indicates that this term promotes better alignment with the performance of downstream tasks while simultaneously minimizing the training loss, thereby improving the generalizability of the representation. Furthermore, we design experiments to examine the empirical performance of Contrast-Reg by formulating the graph contrastive learning framework into four components: a similarity definition, a GNN encoder, a contrastive loss function, and a downstream task. We apply Contrast-Reg to different compositions of these components and achieve superior results across various compositions.

The main contributions of this paper can be summarized as follows:

- (Section 4.1) We identify limitations in existing graph contrastive learning algorithms when applying them to new tasks or data, and adapt the expected calibration error (ECE) to assess the quality of embeddings and their accuracy in downstream tasks.
- (Section 4.2) We propose a novel regularization method, Contrast-Reg, that addresses miscalibration issues, ensuring that minimizing the contrastive loss results in high-quality representations and improved accuracy in downstream tasks.
- (Section 4.2 & Section 5 & Section 6) We provide both theoretical evidence and empirical experiments to support the effectiveness of Contrast-Reg, demonstrating its ability to improve generalizability and achieve superior results across different components of the graph contrastive learning framework.

2 Related Work

Graph representation learning. Many graph representation learning models have been proposed. Factorization-based models Ahmed et al. (2013); Cao et al. (2015); Qiu et al. (2018) factorize an adjacency matrix to obtain node representations. Random walk-based models such as DeepWalk Perozzi et al. (2014) sample node sequences as the input to skip-gram models to compute the representation for each node. Node2vec Grover & Leskovec (2016) balances depth-first and breadth-first random walk when it samples node sequences. HARP Chen et al. (2018) compresses nodes into super-nodes to obtain a hierarchical graph to provide hierarchical information to random walk. GNN models Kipf & Welling (2017); Hamilton et al. (2017); Velickovic et al. (2018); Xu et al. (2019); Qu et al. (2019); Thomas et al. (2023) 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, $\tilde{h}_i^k = aggregate(h_j^{k-1}, j \in neighborhood(i))$, and $h_i^k = combine(\tilde{h}_i^k, h_i^{k-1})$. This work employs GNN models as the backbone and tests the representation across various downstream tasks, such as node classification, link prediction, and graph clustering.

Graph 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 various ways for different types of data and tasks, such as multi-view Tian et al. (2020; 2019), target-to-noise van den Oord et al. (2018); Hénaff et al. (2019), mutual information Belghazi et al. (2018); Hjelm et al. (2019), instance discrimination Wu et al. (2018), context co-occurrence Mikolov et al. (2013), clustering Asano et al. (2020); Caron et al. (2018), multiple data augmentation Chen et al. (2020), known and novel pairs Sun & Li (2023), and contextually relevant Neelakantan et al. (2022). Contrastive learning has been successfully applied to numerous graph representation learning models, such as Perozzi et al. (2014); Grover & Leskovec (2016); Tang et al. (2015); Velickovic et al. (2019); Peng et al. (2020); Hamilton et al. (2017); You et al. (2020); Qiu et al. (2020); Zeng et al. (2021), to task subgraph instance discrimination as a contrastive learning training objective and to leverage contrastive learning to empower graph neural networks in learning node representations. We characterize different types of node-level similarity as follows:

- **Structural similarity**: Structural similarity can be captured from various perspectives. From a graph theory viewpoint, GraphWave Donnat et al. (2018) leverages the diffusion of spectral graph wavelets to capture structural similarity, while struc2vec Ribeiro et al. (2017) uses a hierarchy to measure node similarity at different scales. From an induced subgraph perspective, GCC Qiu et al. (2020) treats the induced subgraphs of the same ego network as similar pairs and those from different ego networks as dissimilar pairs. To capture community structure, vGraph Sun et al. (2019) utilizes the high correlation between community detection and node representations to incorporate more community structure information into node representations. To capture global-local structure, DGI Velickovic et al. (2019) 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 Peng et al. (2020) maximizes the mutual information between node attributes and high-level representations, and Hu et al. (2020b) applies attribute masking to help capture domain-specific knowledge.

Given the above subgraph instance discrimination objective with GNN backbones, NCElossGutmann & Hyvärinen (2012); Dyer (2014); Mnih & Teh (2012); Tian et al. (2020; 2019); van den Oord et al. (2018); Hénaff et al. (2019); Belghazi et al. (2018); Hjelm et al. (2019); Wu et al. (2018); Chen et al. (2020); Yang et al. (2020b) is applied to optimize the model's parameters.

In this work, our focus is on graph contrastive learning. We will demonstrate why graph contrastive learning does not always work in downstream tasks and propose a regularization term to improve the generalizability of various graph contrastive learning algorithms.

Expected Calibration Error. Expected Calibration Error (ECE) Guo et al. (2017) is a metric employed to quantify the calibration between *confidence* (largest predicted probability) and *accuracy* in a model. Calibration refers to the consistency between a model's predicted probabilities and the actual outcomes. When a model is miscalibrated, its generalization to unseen data in supervised learning tasks is likely to be poor Müller et al. (2019); Pereyra et al. (2017); Guo et al. (2017); Zhang et al. (2018). We propose using ECE to evaluate the quality of node embeddings generated by unsupervised graph contrastive learning models. This calibration offers insights into addressing the challenge that applying graph contrastive learning algorithms to downstream tasks does not always yield optimal results.

Regularization for graph representation learning. GraphAT Feng et al. (2019) and BVAT Deng et al. (2019) introduce adversarial perturbations $\frac{\partial f}{\partial x}$ to the input data x as regularizers to obtain more robust models. GraphSGAN Ding et al. (2018) generates fake input data in low-density regions by incorporating a generative adversarial network as a regularizer. P-reg Yang et al. (2020a) leverages the smoothness property in real-world graphs to enhance GNN models. Graphnorm Cai et al. (2021) proposes a novel feature normalization method. Zhou et al. (2021) employs label propagation to adaptively integrate label smoothing into GNN training. Han et al. (2022) use graphons as a surrogate to apply mixup techniques to graph data.

The above regularizers are designed for general representation generalizability, while Contrast-Reg is specifically intended to address the miscalibration problem in the unsupervised graph contrastive learning optimization process and its performance on downstream tasks. It is worth noting that Contrast-Reg could be used in conjunction with the aforementioned regularization techniques.

3 Preliminaries

We begin by introducing the concepts and foundations of graph contrastive learning (GCL). Let a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be denoted, where $\mathcal{V} = v_1, v_2, \cdots, v_n$ and $\mathcal{E}$ represent the vertex set and edge set of $\mathcal{G}$, and the node feature vector of node v_i be x_i. Our objective is to learn node embeddings through unsupervised graph contrastive learning and subsequently apply simple classifiers leveraging these embeddings for various downstream tasks, such as node classification, link prediction, and graph clustering.

Graph contrastive learning Given a graph $\mathcal{G}$ with node features $\mathcal{X} = (x_1, x_2, \cdots, x_n)$, the aim of graph contrastive learning is to train an encoder $f : (\mathcal{G}, \mathcal{X}) \to \mathbb{R}^d$ for all input data points $v_i \in \mathcal{V}$ with node feature vector x_i by constructing positive pairs (v_i, v_i^+) and negative pairs $(v_i, v_{i1}^-, \cdots, v_{iK}^-)$.

The encoder f is typically implemented using Graph Neural Networks (GNNs). Specifically, let h_i represent the output embedding of the encoder f for node v_i, $h_i^{(k)}$ as the embedding at the k-th layer, and $h_i^0 = x_i$. The output of the encoder f is then iteratively defined as:

$$\begin{aligned} m_i^{(k)} &= \text{Aggregate}_k \left(\left\{ h_j^{(k-1)} : v_j \in \mathcal{N}(v_i) \right\} \right) \\ h_i^{(k)} &= \text{RELU} \left(W^k \cdot \text{Update} \left(h_i^{(k-1)}, m_i^{(k)} \right) \right), \end{aligned} \tag{1}$$

where $\mathcal{N}(v_i)$ the set of nodes adjacent to v_i, and Aggregate and Update are the aggregation and update function of GNNs Gilmer et al. (2017). h_i is the last layer of $h_i^{(k)}$ for node v_i.

Leveraging various types of similarity as pseudo subgraph instance discrimination labels, graph contrastive learning constructs positive and negative pairs to train the embedding h_i by optimizing the loss on these pairs. The most commonly employed loss is the NCE loss:

$$\hat{\mathcal{L}}_{nce} = \frac{1}{M} \sum_{i=1}^{M} \left[-\log \sigma(h_i^T h_i^+) + \sum_{k=1}^{K} \log \sigma(h_i^T h_{ij}^-) \right] \tag{2}$$

with M samples $\left(v_i, v_i^+, v_{i1}^-, \cdots, v_{iK}^-\right)_{i=1}^M$ in empirical setting where $\sigma(\cdot)$ is the sigmoid function.

Calibrating graph contrastive learning by the expected calibration error (ECE) Expected Calibration error measures the degree to which the model output probabilities match ground-truth accuracies in supervised tasks Naeini et al. (2015); Guo et al. (2017). It's defined as the expectation of absolute difference between the *largest predicted probability (confidence)* and its corresponding *accuracy*,

$$\text{ECE} = \mathbb{E}_{(v,v') \in S} \left[|p(v, v') - acc(v, v')| \right] \tag{3}$$

In this paper, we extend the use of ECE to evaluate the quality of graph contrastive learning. Specifically, we can compare the predicted probability of a positive pair (i.e., a pair of nodes with the same label) with the true probability that the pair belongs to the same class. We can also compare the predicted probability of a negative pair (i.e., a pair of nodes with different labels) with the true probability that the pair belongs to different classes. By calculating the differences between the predicted and true probabilities for both positive and negative pairs, we can quantify the overall miscalibration of the model and identify areas for improvement. The formal definition is as follows. The largest predicted probability, $p(v, v')$, is determined as

$$p(v, v') = \begin{cases} \sigma(h_{v'} \cdot h_v), & (v, v') \text{ as positive pair} \\ 1 - \sigma(h_{v'} \cdot h_v), & (v, v') \text{ as negative pair} \end{cases} \tag{4}$$

where h_v and $h_{v'}$ are the embeddings for the target node v and the selected sample v', respectively, and $\sigma(\cdot)$ is the sigmoid function. The corresponding accuracy, $acc(v, v')$, is defined as follows:

$$acc(v, v') = \begin{cases} \mathbb{I}(v, v'), & (v, v') \text{ as positive pair} \\ 1 - \mathbb{I}(v, v'), & (v, v') \text{ as negative pair} \end{cases} \tag{5}$$

where $\mathbb{I}$ is an indicator function denoting whether v and v' has the same label, the positive/negative pairs are pseudo-labels for training the contrastive learning algorithm which may not equal to the true label. In this case, when two nodes are selected as positive pairs, $acc(v, v')$ is equal to 1 if v' and v belong to the same class and equal to 0 otherwise; when the two nodes are selected as negative pairs, $acc(v, v') = 1 - \mathbb{I}(v, v')$.

Using the ECE metric in this way allows us to evaluate the quality of the embeddings outputted by the model ($\sigma(h_{v'} \cdot h_v)$) and their accuracy in downstream tasks ($acc(v, v')$), such as node classification. We can improve the general performance and utility of graph contrastive learning algorithms by detecting regions of miscalibration and developing novel techniques for enhancing calibration performance.

4 Methodology

To calibrate the model, we first adapt the expected calibration error (ECE) formula for the graph contrastive learning setting (Section 4.1). In order to ensure that minimizing the contrastive loss leads to high-quality representations that improve accuracy in downstream tasks, we propose a regularization term designed to lower the ECE value and enhance the model's generalizability (Section 4.2). We also provide theoretical evidence to support the effectiveness of the proposed regularization term.

4.1 Empirical Calibration Reveals Limitations of Existing Graph Contrastive Learning Algorithms

To calibrate the existing graph contrastive learning algorithms, we measure the degree of calibration using the expected calibration error (ECE) metric, which is defined by Equation 3, 4, 5:

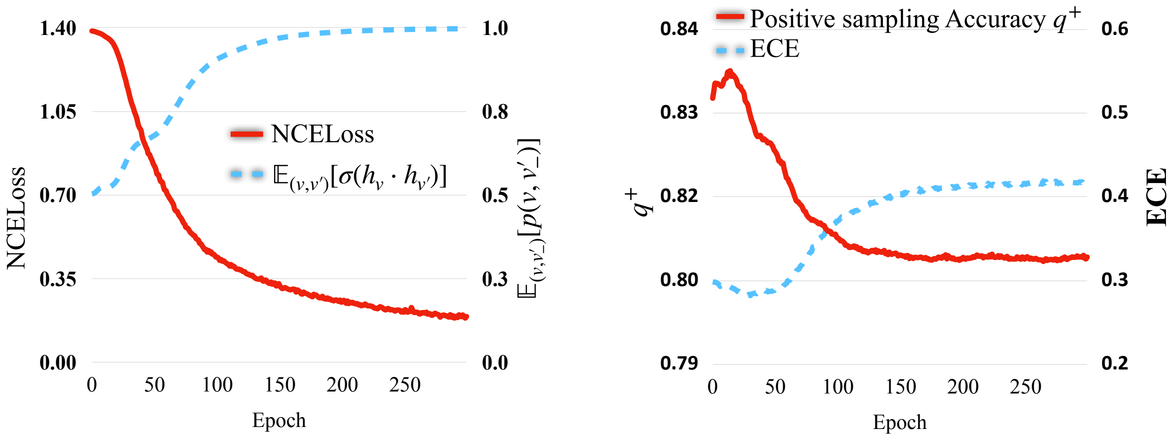
$$\begin{aligned} \text{ECE} = & r^{+} \cdot \Bigg(\Big(1 - \mathbb{E}_{acc(v,v'_{+})=1}[p(v, v'_{+})]\Big)\, q^{+} && (\textit{true positive}) \\ & \quad + \mathbb{E}_{acc(v,v'_{+})=0}[p(v, v'_{+})] \cdot \big(1 - q^{+}\big) \Bigg) && (\textit{false positive}) \\ & + r^{-} \Bigg(\mathbb{E}_{acc(v,v'_{-})=1}[p(v, v'_{-})] \cdot q^{-} && (\textit{false negative}) \\ & \quad + \Big(1 - \mathbb{E}_{acc(v,v'_{-})=0}[p(v, v'_{-})]\Big) \cdot (1 - q^{-}) \Bigg) && (\textit{true negative}), \end{aligned} \tag{6}$$

where q^{+} and q^{-} are the probabilities that the node v' has the same label as node v in positive and negative sampling, respectively; r^{+} and r^{-} are the ratios of sampling positive and negative samples, with $r^{+} + r^{-} = 1$. In this study, we set $r^{+} = r^{-} = 0.5$.

Building upon this formulation, we propose the following claim that takes into account the positive and negative pair construction assumptions to analyze the potential issues that lead to the miscalibration between the decrease in graph contrastive learning training loss and the degradation of downstream task performance.

Claim 4.1. *Under the assumption that negative sampling is uniformly sampled, and positive sampling is sampled based on the calculated distance between pairwise embeddings, ECE is positively correlated with the expectation of the prediction value for randomly sampled pairs* $\mathbb{E}_{v,v'}[\sigma(h_v \cdot h_{v'})]$*, and negatively correlated with the probability* q^{+} *of* v'_{+} *having the same label as* v *in positive sampling.*

We provide a thorough analysis of this claim in Appendix A.1. To investigate the changes in these two factors and the ECE value over the process of minimizing the NCELoss (Equation 2), we conduct an illustrative experiment with an existing graph contrastive learning algorithm and present the relevant values in Figure 1. In Figure 1a, we show that as the NCELoss (red solid line) is minimized over the epochs, the expectation of the prediction value for randomly sampled pairs $\mathbb{E}(v, v')[\sigma(h_v \cdot hv')]$ (blue dashed line) increases. Moreover, Figure 1b demonstrates the model's challenge in identifying genuine positive pairs, with the probability q^{+} initially increasing before gradually declining (red solid line), and the ECE value first decreasing before rising (blue dashed line). These findings suggest that, as the epochs progress, merely reducing the NCELoss is



(a) NCELoss and $\mathbb{E}_{(v,v')}[\sigma(h_v \cdot h_{v'})]$ over epochs

(b) ECE and q^+ over epochs

Figure 1: Calibrating Graph Contrastive Learning

insufficient to enhance the accuracy of downstream tasks. Our empirical calibration shows that the model learns certain spurious features that reduce the loss but are ultimately harmful to downstream tasks when solely minimizing the NCE loss. The above analysis emphasizes the need for graph contrastive learning algorithms to effectively account for the relationships between learned embeddings and the accuracy of downstream tasks. In the next section, we present our approach to mitigate the possible risks associated with spurious feature learning in graph contrastive learning algorithms and show its effectiveness through a number of experiments.

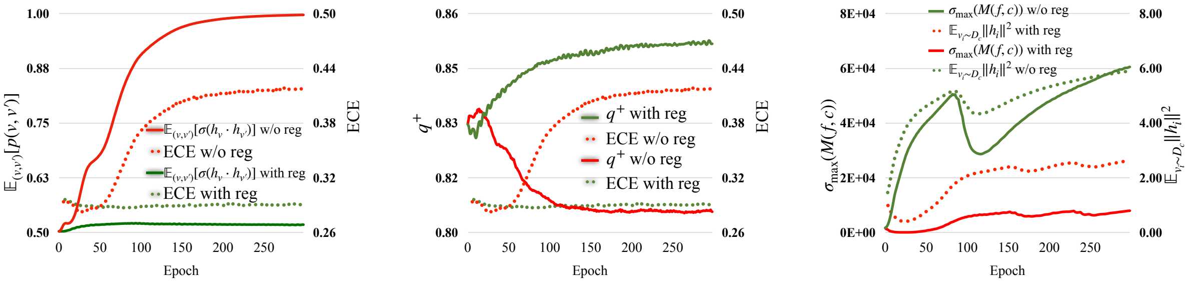
4.2 Proposed Regularization Term: Contrast-Reg

To ensure that minimizing the NCE loss aligns with downstream task accuracy, we propose a contrastive regularization term, denoted as **Contrast-Reg**, given by:

$$\mathcal{L}_{reg} = -\mathop{\mathbb{E}}_{h,\tilde{h}}\left[\log\sigma(h_i^T W\mathbf{r}) + \log\sigma(-\tilde{h}_i^T W\mathbf{r})\right], \tag{7}$$

where $\mathbf{r}$ is a random vector uniformly sampled from $(0,1]$, W is a trainable parameter, and $\tilde{h}$ is the noisy feature generated by different data augmentation techniques, such as those used in Chen et al. (2020); Velickovic et al. (2019). In Section 5, we will discuss how we calculate the noisy features in the GNN setting. In the following, we will introduce the impact of the Contrast-Reg on the ECE value and generalizability.

ECE decreases by incorporating Contrast-Reg We conducted an empirical study to examine the impact of Contrast-Reg on the ECE value and investigate the two factors that cause miscalibration. Figure 2a shows that the expectation of the prediction value for randomly sampled pairs $\mathbb{E}_{(v,v')}[\sigma(h_v \cdot h_{v'})]$ with Contrast-Reg increased much more slowly compared to the vanilla NCELoss (green solid line compared to the red solid line). Figure 2b presents the changes in positive sampling accuracy q^+ over the epochs of the vanilla NCELoss and NCELoss with Contrast-Reg, where Contrast-Reg helps q^+ increase, while q^+ trained by vanilla NCELoss decreased after the initial increases. These two factors contribute to the different ECE changes in the vanilla NCELoss and NCELoss with Contrast-Reg. The red dashed line indicates that the ECE value increases after the initial decreases, while the green dashed line shows that the ECE value decreases slightly. The comparison demonstrates that applying Contrast-Reg alleviates the miscalibration in representation learning, ensuring that minimizing the contrastive loss results in high-quality representations with increased accuracy in downstream tasks, rather than overfitting certain spurious features. We will provide a detailed comparison of the impact on the ECE value between models with and without Contrast-Reg in Appendix A.2 across different datasets.



(a) Influence of Contrast-Reg on ECE and $\mathbb{E}_{(v,v')}[\sigma(h_v \cdot h_{v'})]$

(b) Influence of Contrast-Reg on ECE and q^+

(c) Influence of Contrast-Reg on $\sigma_{\max}(M(f,c))$ and $\mathbb{E}_{v_i \sim D_c}\|h_i\|^2$

Figure 2: Effects of Contrast-Reg

Generalizability improves by incorporating Contrast-Reg In Section 6.3, we present empirical evidence supporting the effectiveness of Contrast-Reg's ability to enhance generalizability through an ablation study. Furthermore, we analyze the impact of Contrast-Reg on the generalizability of graph contrastive learning algorithms using the theoretical analysis tool developed by Saunshi et al. (2019). In Theorem 1, we offer performance guarantees for the learned graph embeddings outputted by the GNN function class $\mathcal{F} = f$ with the unsupervised loss function $\mathcal{L}nce$ on the downstream average classification task $\mathcal{L}sup^{\mu}(\hat{f})$. Detailed settings can be found in Appendix A.4. Assume that f is bounded, i.e., $\|h_i\| \leq R$ with $R > 0$. Let c, c' be two classes sampled independently from latent classes $\mathcal{C}$ with distribution ρ. 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. Let $\mathcal{L}_{nce}^{\neq}(f)$ be the NCELoss when negative samples come from different classes. We have the following theorem.

Theorem 1. *$\forall f \in \mathcal{F}$, with probability at least $1 - \delta$,*

$$\mathcal{L}_{sup}^{\mu}(\hat{f}) \leq \mathcal{L}_{nce}^{\neq}(f) + \beta s(f) + \eta Gen_M, \tag{8}$$

where $Gen_M = \frac{8R\mathcal{R}_{\mathcal{S}}(\mathcal{F})}{M} - 8\log(\sigma(-R^2))\sqrt{\frac{\log\frac{4}{\delta}}{2M}} = O\left(R\frac{\mathcal{R}_{\mathcal{S}}(\mathcal{F})}{M} + R^2\sqrt{\frac{\log\frac{1}{\delta}}{M}}\right)$, $\beta = \frac{\tau}{1-\tau}$, $\eta = \frac{1}{1-\tau}$, *and* $s(f) = 4\sqrt{\mathbb{E}_{(v_i,v_j)\sim\mathcal{D}_{sim}(v_i,v_j)}\left[(h_i^T h_i)^2\right]}$,

In Equation (8), Gen_M represents the generalization error in terms of *Rademacher complexity* and will converge to 0 when the encoder function f is bounded and the number of samples M is sufficiently large. The theorem above indicates that only when $\beta s(f) + \eta Gen_M$ converges to 0 as M increases, will the encoder $\hat{f} = \arg\min_{f \in \mathcal{F}} \hat{\mathcal{L}}_{nce}$ selected perform well in downstream tasks.

Next, we will analyze the impact of Contrast-Reg on the condition in the aforementioned theorem, $\beta s(f) + \eta Gen_M$, by reformulating $s(f)$ as follows:

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

where $M(f,c) \coloneqq \mathbb{E}_{v_i\sim\mathcal{D}_c}\left[h_i h_i^T\right]$.

The advantage of incorporating Contrast-Reg into graph contrastive learning lies in its ability to reduce both the largest singular value of matrix $M(f,c)$ ($\sigma_{\max}(M(f,c))$) and the expectation of the embedding norm within the same class, $\mathbb{E}_{v_i\sim D_c}[\|h_i\|^2]$ (as illustrated in Figure 2c). This reduction leads to a decrease in the upper bound of $s(f)$, ultimately yielding a lower value for the term $\beta s(f) + \eta Gen_M$ compared to vanilla graph contrastive learning. The reduction in this term promotes better alignment with the performance of downstream tasks while simultaneously minimizing the training loss. Further experiments on the generalizability of the Contrast-Reg approach can be found in Section 6.3, and a more detailed explanation on lowering the term $s(f)$ is provided in Appendix A.3.

5 A Contrastive GNN Framework

In this section, we initially introduce the graph contrastive framework in Algorithm 1. Following that, we provide two illustrative graph contrastive learning algorithms, each accompanied by its respective similarity definition. In Section 6, we will conduct experiments using the graph contrastive learning framework to showcase Contrast-Reg's capability in addressing the miscalibration problem and enhancing various components within the general graph contrastive learning framework.

The framework involves training a GNN model f for e epochs using a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and node attributes $\mathcal{X}$. Node representations obtained through f can be used as inputs for downstream tasks, and we employ NCEloss as the contrastive loss in our framework. For each training epoch, we first select a seed node set $\mathcal{C}$ for computing NCEloss by invoking the *SeedSelect* function (Line 3). We then call the *Constrast* function (Line 4) to construct a positive sample and a negative sample for each node in $\mathcal{C}$. The *Constrast* function returns a set $\mathcal{P}$ of 3-tuples consisting of the representations for the seed nodes, the positive samples, and the negative samples. We compute the training loss by adding NCEloss on $\mathcal{P}$ and the regularization loss calculated by Contrast-Reg (Line 5) and update f by back-propagation (Line 6). As discussed in Section 4, Contrast-Reg necessitates noisy features for contrastive regularization. In our contrastive GNN framework, we generate these noisy features by shuffling node features among nodes, following the corruption function employed in Velickovic et al. (2019). Different *SeedSelect* and *Contrast* functions are designed for various node similarity definitions to select seed nodes that yield effective training results and generate appropriate contrastive pairs for these seed nodes. In the following section, we present two examples of contrastive GNN models for structure and attribute similarities, respectively. These examples are also utilized in our experimental evaluation in Section 6.

```
Algorithm 1: Contrastive GNN Framework
Input: Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, node attributes $\mathcal{X}$, 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     $\mathcal{C}$=SeedSelect($\mathcal{G}$, $\mathcal{X}$, $f$, $epoch$);
4     $\mathcal{P}$=Contrast($\mathcal{C}$, $\mathcal{G}$, $\mathcal{X}$, $f$);
5     loss = NCEloss($\mathcal{P}$) + Contrast-Reg($\mathcal{G}$, $\mathcal{X}$, $f$);
6     Back-propagation and update $f$;
7 end
```

```
Algorithm 2: ML
Parameter: Parameters of an (additional) GNN
           layer $g$.
Function Contrast($\mathcal{C}$, $\mathcal{G}$, $\mathcal{X}$, $f$):
    Let $g(x_i)$ be the representation of $x_i$ by
      stacking $g$ upon $f$;
    Randomly pick a negative node $x_i^-$ from $\mathcal{V}$
      for each $x_i \in \mathcal{C}$;
    return $\left\{(g(x_i), f(x_i), f(x_i^-))\right\}_{x_i \in \mathcal{G}}$;
end
Function SeedSelect($\mathcal{G}$, $\mathcal{X}$, $f$, $epoch$):
    return $\mathcal{V}$;
end
```

5.1 Attribute Similarity

Models adopting attribute similarity assume that nodes with similar attributes should have similar representations, ensuring that the attribute information is preserved. Hjelm et al. (2019); Peng et al. (2020) proposed contrastive pair designs to maximize the mutual information between low-level representations (input features) and high-level representations (learned representations). Algorithm 2 presents our model, ML, which adapts their multi-level representation design into our contrastive GNN framework. In Algorithm 2, *SeedSelect* selects all nodes in a graph as seeds. *Contrast* uses the node x_i itself as the positive node for each seed node x_i. However, in the returned 3-tuple, the representation of x_i as the seed node differs 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 g upon f. For negative nodes, *Contrast* randomly samples a node in $\mathcal{V}$ for each seed node.

5.2 Structural Similarity

We provide an example model (*LC*) that captures the community structure inherent in graph data Newman (2006). Since clustering is a common and effective method for detecting communities in a graph, we con-

Algorithm 3: LC

```
Hyperparameter: $R$: curriculum update epochs; $k$: the number of candidate positive samples for seed node;
1  Function Contrast($\mathcal{C}$, $\mathcal{G}$, $\mathcal{X}$, $f$):
2      For $x_i \in \mathcal{C}$, let $\mathcal{N}_i^+$ 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 $\mathcal{N}_i^+$ for each $x_i \in \mathcal{C}$;
4      Randomly pick one negative node $x_i^-$ from $\mathcal{V}$ for each $x_i \in \mathcal{C}$;
5      return $\left\{(f(x_i), f(x_i^+), f(x_i^-))\right\}_{x_i \in \mathcal{C}}$;
6  end
7  Function SeedSelect($\mathcal{G}$, $\mathcal{X}$, $f$, $epoch$):
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
```

duct clustering in the node representation space to capture community structures. LC borrows the design from Huang et al. (2019) and implements local clustering using the *Contrast* function, along with curriculum learning through the *SeedSelect* function. We note that other methods, such as global clustering Caron et al. (2018) and instance discrimination Wu et al. (2018), can also be adapted into our contrastive GNN framework with different implementations of *Contrast* and *SeedSelect*. Algorithm 3 demonstrates 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 $\mathcal{V}$ (Lines 2-4). *SeedSelect* selects nodes with the smallest entropy to avoid high randomness and uncertainty at the beginning of the training process. For every R epochs, *SeedSelect* gradually adds more nodes with larger entropy to be computed in the contrastive loss as the epochs progress (Lines 11-13).

6 Experimental Results

We begin by introducing the experimental settings in Section 6.1. Section 6.2 presents the main results across various downstream tasks. Moreover, we assess the benefits of Contrast-Reg through ablation studies.

6.1 Experiment Settings

Downstream tasks We commence our experimentation on three distinct downstream tasks, namely node classification, graph clustering, and link prediction. The experimental procedure consists of two stages: first, we utilize positive and negative contrastive pairs to train the GNN models in an unsupervised manner, obtaining the node embeddings. Subsequently, we apply these embeddings to the downstream tasks by integrating additional straightforward models. For example, we employ multiclass logistic regression for the node classification, k-means for graph clustering, and a single MLP layer for link prediction. Moreover, we conduct experiments in the pretrain-finetune paradigm, as this approach constitutes a significant component of graph contrastive learning Qiu et al. (2020). We first employ a large graph to train the GNN models, followed by finetuning such models and train a simple downstream classifier on a separate graph.

Datasets The datasets we employed encompass citation networks, web graphs, co-purchase networks, and social networks. Comprehensive statistics for these datasets can be found in Appendix B.

Similarity definition We evaluate our proposed Contrast-Reg using two similarity definitions: (a) Algorithm 3 captures *structure similarity* and is denoted as **ours (LC)**, while (b) Algorithm 2 captures *attribute similarity* and is denoted as **ours (ML)**.

Table 1: 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	**71.74**±0.29	**72.40**±0.95	79.82±2.04	**88.75**±1.99	75.64±0.21	94.02±0.05
node2vec	71.07±0.91	47.37±0.95	66.34±1.40	**70.07**±0.13	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	**77.00**±0.21	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)	**82.33**±0.41	**72.88**±0.39	**79.33**±0.59	69.94±0.11	**69.19**±1.13	**81.98**±1.52	**87.59**±1.50	**76.96**±0.34	**94.43**±0.03
ours (ML)	**82.65**±0.57	**72.98**±0.41	**80.10**±1.04	**70.05**±0.09	67.20±0.96	**82.11**±1.47	86.78±1.70	76.27±0.20	**94.38**±0.04

Table 2: 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	**71.81**±1.01	54.90±0.66	**69.88**±0.90	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	**56.01**±0.64	68.11±1.32	**69.17**±0.43	**44.47**±0.46	**64.74**±0.41	**53.13**±1.01	**51.81**±0.57	**46.11**±0.93

Training Details We employed full-batch training for Cora, Citeseer, Pubmed, ogbn-arxiv, Wiki, Computers, and Photo, while utilizing stochastic mini-batch training for Reddit and ogbn-products. For Cora, Citeseer, Pubmed, ogbn-arxiv, ogbn-products, and Reddit, we adhered to the standard dataset splits and conducted 10 different runs with fixed random seeds ranging from 0 to 9. For Computers, Photo, and Wiki, we randomly divided the train/validation/test sets, allocating 20/30/all remaining nodes per class, in accordance with the recommendations in Shchur et al. (2018). We measured performance across 25 (5×5) different runs, comprising 5 random splits and 5 fixed-seed runs (from 0 to 4) for each random split. The hyperparameter configurations can be found in Appendix B.

6.2 Main Results

Our primary results involve comparing our proposed Contrast-Reg, along with the chosen similarity definitions, against state-of-the-art algorithms across various downstream tasks.

6.2.1 Node Classification

We evaluated node classification performance on all datasets, utilizing both full-batch training and stochastic mini-batch training. Our methods were compared with DGI Velickovic et al. (2019), GMI Peng et al. (2020), node2vec Grover & Leskovec (2016), and supervised GCN Kipf & Welling (2017). DGI and GMI represent state-of-the-art algorithms in unsupervised graph contrastive learning. Node2vec is an exemplary algorithm for random walk-based graph representation algorithms Grover & Leskovec (2016); Tang et al. (2015); Perozzi et al. (2014), while GCN is a classic semi-supervised GNN model. We report the performance of ours (LC) and ours (ML), both utilizing Contrast-Reg. For full-batch training, the encoder is GCN, whereas for stochastic training, the encoder is GraphSage Hamilton et al. (2017) with GCN-aggregation. The encoder settings are consistent with those in DGI and GMI. Table 1 presents the node classification accuracy, including standard deviation. Our results show that our algorithms outperform in the majority of cases for both full-batch training (on Cora, Citeseer, Pubmed, Computers, Photo, and Wiki) and stochastic training (on Reddit and Ogbn-products). Remarkably, our unsupervised algorithms can even surpass the performance of the supervised GCN.

6.2.2 Graph Clustering

We assessed clustering performance using three metrics: accuracy (Acc), normalized mutual information (NMI), and F1-macro (F1), following the work of Xia et al. (2014). Higher values indicate better clustering performance. We compared our methods with DGI, node2vec, GMI, and AGC Zhang et al. (2019) on the

Table 3: Downstream task: link prediction

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

Table 4: Pretraining

Algorithm	Reddit	ogbn-products
No pretraining	90.44±1.62	84.69±0.79
DGI	92.09±1.05	86.37±0.19
GMI	92.13±1.16	86.14±0.16
ours (ML)	92.18±0.97	86.28±0.20
ours (LC)	**92.52**±0.55	**86.45**±0.13

Cora, Citeseer, and Wiki datasets. AGC is a state-of-the-art graph clustering algorithm that leverages high-order graph convolution for attribute graph clustering. For all models and datasets, we employed k-means to cluster both the labels and representations of nodes. The clustering results of labels were considered as the ground truth. To reduce dimensionality, we applied PCA to the representations before using k-means, since high dimensionality can negatively impact clustering Chen (2009). The random seed setting for model training was consistent with that in the node classification task. To minimize randomness, we set the clustering random seed from 0 to 4 and computed the average result for each learned representation. Table 2 presents improved results with and without PCA for each cell. Our algorithms, particularly ours (ML), exhibited superior performance in all cases, demonstrating the effectiveness of Contrast-Reg. It is worth noting that the superior results of ours (ML) compared to ours (LC) suggest that attributes play a crucial role in clustering, as graph clustering is applied to attribute graphs.

6.2.3 Link Prediction

In order to circumvent the data linkage issue in link prediction, we employed an inductive setting for graph representation learning. We randomly extracted induced subgraphs (comprising 85% of the edges) from each original graph for training both the representation learning model and the link predictor, while reserving the remaining edges for validation and testing (10% for the test edge set and 5% for the validation edge set). We assessed performance across 25 (5x5) different runs, utilizing a fixed-seed random split scheme with five distinct induced subgraphs and five fixed-seed runs (ranging from 0 to 4). We compared our model with DGI, GMI, node2vec, and unsupervised GCN (GCN-neg in Table 3) on the Cora, Citeseer, Pubmed, and Wiki datasets. It is important to note that we did not include the ML model in this experiment, as it primarily focuses on node attributes. The results presented in Table 3 demonstrate that our algorithms surpass the performance of state-of-the-art methods.

6.2.4 Pretraining

We further assessed the performance of Contrast-Reg in the context of the pretrain-finetune paradigm. For the Reddit dataset, we naturally partitioned the data by time, pretraining the models using the first 20 days. We generated an induced subgraph based on the pretraining nodes and divided the remaining data into three parts: the first part produced a new subgraph for fine-tuning the pre-trained model and training the classifier, while the second and third parts were designated for validation and testing. For the ogbn-products dataset, we split the data according to node ID, pretraining the models using a subgraph generated by the initial 70% of the nodes. The data splitting scheme for the remaining data mirrored that of the Reddit dataset. We conducted baseline experiments on DGI and GMI, employing the same GraphSAGE with GCN-aggregation encoder as in our model. Table 4 reveals that pretraining the model facilitates convergence to a

Table 5: Contrastive learning with and w/o Contrast-Reg

Algorithm	Cora	Wiki	Computers	Reddit
ML	73.22±0.77	58.70±1.51	77.08±2.48	94.33±0.07
ML+ℓ_2-normalization	80.14±0.92	61.83±1.36	80.80±1.45	94.07±0.19
ML+Weight-decay	81.65±0.42	63.67±1.47	79.09±0.32	94.34±0.06
ML+Contrast-Reg	**82.65**±0.57	**67.20**±0.96	**82.11**±1.47	**94.38**±0.04
LC	79.73±0.75	65.14±1.48	79.80±1.49	94.42±0.03
LC+ℓ_2-normalization	81.09±0.59	63.96±0.64	80.73±1.36	94.20±0.06
LC+Weight-decay	81.94±0.44	65.17±1.34	81.89±1.58	**94.44**±0.03
LC+Contrast-Reg	**82.33**±0.41	**69.19**±1.13	**81.98**±1.52	**94.43**±0.03

more robust representation model with reduced variance, and Contrast-Reg can enhance the transferability of the pre-trained model.

6.3 Discussion of Contrast-Reg

To evaluate the benefits of Contrast-Reg, we conduct experiments comparing it to two approaches, specifically, ℓ_2-normalization and weight decay, on four networks from different domains, with a focus on node classification task performance. It is important to note that some regularization techniques, such as those mentioned in Section 2, are not explicitly designed to address the miscalibration problem. ℓ_2-normalization Chen et al. (2020) mitigates the potential risk of the expectation of the prediction value for randomly sampled pairs $\mathbb{E}_{(v,v')}[\sigma(v \cdot v')]$ exploding by explicitly eliminating the embedding norm for each node's embeddings to tackle the miscalibration problem, while weight decay strives to achieve the same result by implicitly restricting the gradient descent step length. Table 5 presents a comparison of the accuracy achieved by different contrastive learning algorithms, **ML** and **LC**, when incorporating ℓ_2-normalization, weight decay, and Contrast-Reg, as opposed to using the vanilla algorithms. The results indicate that integrating ℓ_2-normalization, weight decay, and Contrast-Reg into graph contrastive learning algorithms improves the accuracy of downstream tasks, suggesting that addressing miscalibration enhances the generalization of learned embeddings for downstream tasks. However, the performance gains provided by Contrast-Reg exceed those of ℓ_2-normalization and weight decay. This implies that while alternative algorithms exist to address miscalibration, Contrast-Reg emerges as the most effective method for improving the generalization of learned embeddings for downstream tasks. It is crucial to acknowledge the existence of other algorithms that may also contribute to mitigating the miscalibration problem, and further research is needed to explore and compare their effectiveness.

Appendix A.9 presents the ablation study of Contrast-Reg on the GAT backbone, illustrating that Contrast-Reg consistently enhances graph contrastive learning performance across various backbones. In conclusion, based on the experimental results, we determine that Contrast-Reg serves as an effective regularization method for a general graph contrastive learning framework, encompassing similarity definition, GNN encoder backbone, and downstream tasks.

7 Conclusions

In conclusion, graph contrastive learning algorithms have shown great potential in various applications, such as node classification, link prediction, and graph clustering. However, the effectiveness of these algorithms in new tasks or data can be inconsistent. By adapting the expected calibration error (ECE) to the graph contrastive learning framework, we analyzed the shortcomings of existing algorithms and addressed the issue of miscalibration. Our novel regularization method, Contrast-Reg, significantly enhances the quality of embeddings and their performance in downstream tasks. Through theoretical evidence and empirical experiments, we have demonstrated the effectiveness of Contrast-Reg in improving the generalizability of graph contrastive learning algorithms. This research paves the way for the development of more robust and reliable graph representation learning techniques, ultimately benefiting a wide range of applications across various domains.

References

Amr Ahmed, Nino Shervashidze, Shravan M. Narayanamurthy, Vanja Josifovski, and Alexander J. Smola. Distributed large-scale natural graph factorization. In *WWW '13*, pp. 37–48, 2013.

Yuki M. Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *ICLR '20*, 2020.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, R. Devon Hjelm, and Aaron C. Courville. Mutual information neural estimation. In *ICML '18*, pp. 530–539, 2018.

Stephen P. Boyd and Lieven Vandenberghe. *Convex Optimization.* 2014.

Tianle Cai, Shengjie Luo, Keyulu Xu, Di He, Tie-Yan Liu, and Liwei Wang. Graphnorm: A principled approach to accelerating graph neural network training. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 1204–1215. PMLR, 2021. URL `http://proceedings.mlr.press/v139/cai21e.html`.

Shaosheng Cao, Wei Lu, and Qiongkai Xu. Grarep: Learning graph representations with global structural information. In *CIKM '15*, pp. 891–900, 2015.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV '18*, pp. 139–156, 2018.

Haochen Chen, Bryan Perozzi, Yifan Hu, and Steven Skiena. Harp: Hierarchical representation learning for networks. In *(AAAI) '18, (IAAI) '18, and (EAAI) '18*, pp. 2127–2134, 2018.

Lei Chen. Curse of dimensionality. In *Encyclopedia of Database Systems*, pp. 545–546. 2009.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. *CoRR*, abs/2002.05709, 2020.

Zhijie Deng, Yinpeng Dong, and Jun Zhu. Batch virtual adversarial training for graph convolutional networks. *CoRR*, abs/1902.09192, 2019.

Ming Ding, Jie Tang, and Jie Zhang. Semi-supervised learning on graphs with generative adversarial nets. In *CIKM '18*, pp. 913–922, 2018.

Claire Donnat, Marinka Zitnik, David Hallac, and Jure Leskovec. Learning structural node embeddings via diffusion wavelets. In *KDD '18*, pp. 1320–1329, 2018.

Chris Dyer. Notes on noise contrastive estimation and negative sampling. 2014.

Fuli Feng, Xiangnan He, Jie Tang, and Tat-Seng Chua. Graph adversarial training: Dynamically regularizing based on graph structure. *CoRR*, abs/1902.08226, 2019.

Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. *CoRR*, abs/1903.02428, 2019.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *ICML '17*, 2017.

Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *KDD '16*, pp. 855–864, 2016.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1321–1330. PMLR, 2017. URL `http://proceedings.mlr.press/v70/guo17a.html`.

Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *J. Mach. Learn. Res.*, 13:307–361, 2012.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NeurIPS '17*, pp. 1024–1034, 2017.

Xiaotian Han, Zhimeng Jiang, Ninghao Liu, and Xia Hu. G-mixup: Graph data augmentation for graph classification. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 8230–8248. PMLR, 2022. URL `https://proceedings.mlr.press/v162/han22c.html`.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR '20*, pp. 9726–9735, 2020.

Olivier J. Hénaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, S. M. Ali Eslami, and Aäron van den Oord. Data-efficient image recognition with contrastive predictive coding. *CoRR*, abs/1905.09272, 2019.

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 '19*, 2019.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *CoRR*, abs/2005.00687, 2020a.

Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay S. Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. In *ICLR '20*, 2020b.

Jiabo Huang, Qi Dong, Shaogang Gong, and Xiatian Zhu. Unsupervised deep learning by neighbourhood discovery. In *ICML '19*, pp. 2849–2858, 2019.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR '17*, 2017.

Andreas Maurer. A vector-contraction inequality for rademacher complexities. In Ronald Ortner, Hans Ulrich Simon, and Sandra Zilles (eds.), *ALT '16*, pp. 3–17, 2016.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *NeurIPS '13*, pp. 3111–3119, 2013.

Andriy Mnih and Yee Whye Teh. A fast and simple algorithm for training neural probabilistic language models. In *ICML '12*, 2012.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. The MIT Press, 2nd edition, 2018. ISBN 0262039400.

Rafael Müller, Simon Kornblith, and Geoffrey E. Hinton. When does label smoothing help? In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 4696–4705, 2019. URL `https://proceedings.neurips.cc/paper/2019/hash/f1748d6b0fd9d439f71450117eba2725-Abstract.html`.

Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In Blai Bonet and Sven Koenig (eds.), *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, pp. 2901–2907. AAAI Press, 2015.

Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, Johannes Heidecke, Pranav Shyam, Boris Power, Tyna Eloundou Nekoul, Girish Sastry, Gretchen Krueger, David Schnurr, Felipe Petroski Such, Kenny Hsu, Madeleine Thompson, Tabarak Khan, Toki Sherbakov, Joanne Jang, Peter Welinder, and Lilian Weng. Text and code embeddings by contrastive pre-training. *CoRR*, abs/2201.10005, 2022. URL `https://arxiv.org/abs/2201.10005`.

M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006.

Zhen Peng, Wenbing Huang, Minnan Luo, Qinghua Zheng, Yu Rong, Tingyang Xu, and Junzhou Huang. Graph representation learning via graphical mutual information maximization. In *WWW '20*, pp. 259–270, 2020.

Gabriel Pereyra, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey E. Hinton. Regularizing neural networks by penalizing confident output distributions. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net, 2017. URL `https://openreview.net/forum?id=HyhbYrGYe`.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: online learning of social representations. In *KDD '14*, pp. 701–710, 2014.

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*, pp. 459–467. ACM, 2018.

Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. GCC: graph contrastive coding for graph neural network pre-training. In *KDD '20*, pp. 1150–1160, 2020.

Meng Qu, Yoshua Bengio, and Jian Tang. GMNN: graph markov neural networks. In *ICML '19*, volume 97, pp. 5241–5250, 2019.

Leonardo Filipe Rodrigues Ribeiro, Pedro H. P. Saverese, and Daniel R. Figueiredo. *struc2vec*: Learning node representations from structural identity. In *KDD '17*, pp. 385–394, 2017.

Nikunj Saunshi, Orestis Plevrakis, Sanjeev Arora, Mikhail Khodak, and Hrishikesh Khandeparkar. A theoretical analysis of contrastive unsupervised representation learning. In *ICML '19*, pp. 5628–5637, 2019.

Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation. *CoRR*, abs/1811.05868, 2018.

Fan-Yun Sun, Meng Qu, Jordan Hoffmann, Chin-Wei Huang, and Jian Tang. vgraph: A generative model for joint community detection and node representation learning. In *NeurIPS '19*, pp. 512–522, 2019.

Yiyou Sun and Yixuan Li. Opencon: Open-world contrastive learning. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL `https://openreview.net/forum?id=2wWJxtpFer`.

Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. LINE: large-scale information network embedding. In *WWW '15*, pp. 1067–1077, 2015.

Josephine Thomas, Alice Moallemy-Oureh, Silvia Beddar-Wiesing, and Clara Holzhüter. Graph neural networks designed for different graph types: A survey. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL `https://openreview.net/forum?id=h4BYtZ79uy`.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *CoRR*, abs/1906.05849, 2019.

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning. *CoRR*, abs/2005.10243, 2020.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *ICLR '18*, 2018.

Petar Velickovic, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R. Devon Hjelm. Deep graph infomax. In *ICLR '19*, 2019.

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. Deep graph library: A graph-centric, highly-performant package for graph neural networks. *arXiv preprint arXiv:1909.01315*, 2019.

Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance-level discrimination. *CoRR*, abs/1805.01978, 2018.

Rongkai Xia, Yan Pan, Lei Du, and Jian Yin. Robust multi-view spectral clustering via low-rank and sparse decomposition. In *AAAI '14*, pp. 2149–2155, 2014.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *ICLR '19*, 2019.

Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Y. Chang. Network representation learning with rich text information. In *IJCAI '15*, pp. 2111–2117, 2015.

Han Yang, Kaili Ma, and James Cheng. Rethinking graph regularization for graph neural networks, 2020a.

Zhen Yang, Ming Ding, Chang Zhou, Hongxia Yang, Jingren Zhou, and Jie Tang. Understanding negative sampling in graph representation learning. In *KDD '20*, pp. 1666–1676, 2020b.

Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. In *ICML '16*, volume 48 of *JMLR Workshop and Conference Proceedings*, pp. 40–48, 2016.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL `https://proceedings.neurips.cc/paper/2020/hash/3fe230348e9a12c13120749e3f9fa4cd-Abstract.html`.

Liang Zeng, Jin Xu, Zijun Yao, Yanqiao Zhu, and Jian Li. Graph symbiosis learning. *CoRR*, abs/2106.05455, 2021. URL `https://arxiv.org/abs/2106.05455`.

Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL `https://openreview.net/forum?id=r1Ddp1-Rb`.

Xiaotong Zhang, Han Liu, Qimai Li, and Xiao-Ming Wu. Attributed graph clustering via adaptive graph convolution. In *IJCAI '19*, pp. 4327–4333, 2019.

Kaixiong Zhou, Ninghao Liu, Fan Yang, Zirui Liu, Rui Chen, Li Li, Soo-Hyun Choi, and Xia Hu. Adaptive label smoothing to regularize large-scale graph training. *CoRR*, abs/2108.13555, 2021. URL `https://arxiv.org/abs/2108.13555`.

Table 6: Impact of Contrast-Reg on ECE Value

	Cora	Citeseer	Pubmed
w/o reg (ML)	0.477	0.540	0.399
with reg (ML)	**0.413**	**0.525**	**0.273**
w/o reg (LC)	0.477	0.537	0.416
with reg (LC)	**0.437**	**0.524**	**0.274**

A Appendix

A.1 Calibrating the Existing Graph Contrastive Learning Algorithms

Considering that positive sampling is based on the calculated distance between pairwise embeddings, we can express the following relationship:

$$\mathbb{E}_{acc(v,v'_+)=1}[p(v,v'_+)] = \mathbb{E}_{acc(v,v'_+)=0}[p(v,v'_+)] = \mathbb{E}_{\text{topk } \sigma(h_v,h_{v'})}[\sigma(h_v \cdot h_{v'})] \tag{9}$$

Here, $\mathbb{E}_{\text{topk } \sigma(h_v \cdot h_{v'})}[\sigma(h_v \cdot h_{v'})]$ represents the expectation of the top k pairs (v, v'). When the loss converges, $\mathbb{E}_{\text{topk } \sigma(h_v \cdot h_{v'})}[\sigma(h_v \cdot h_{v'})] \to 1$.

Furthermore, by considering the definition in Equation 4 and the fact that negative samples are uniformly sampled, we can write:

$$\mathbb{E}_{acc(v,v'_-)=0}[p(v,v'_-)] = \mathbb{E}_{acc(v,v'_-)=1}[p(v,v'_-)] = 1 - \mathbb{E}_{(v,v')}[\sigma(v,v')]. \tag{10}$$

Thus, Equation 6 can be reformulated as:

$$\begin{aligned}\text{ECE} = {} & r^+(1 - 2\mathbb{E}_{\text{topk } \sigma(h_v \cdot h_{v'})}[\sigma(h_v \cdot h_{v'})])q^+ + r^-(1 - 2q^-)\mathbb{E}[\sigma(h_v \cdot h_{v'})] + r^-q^- \\ & + r^+\mathbb{E}_{\text{topk } \sigma(h_v \cdot h_{v'})}[\sigma(h_v \cdot h'_v)].\end{aligned} \tag{11}$$

When the loss converges, $\mathbb{E}_{\text{topk } \sigma(h_v \cdot h_{v'})}[\sigma(h_v \cdot h'_v)] \to 1$, indicating that the ECE value is negatively correlated with the probability q^+ of v'_+. In addition, negative samples are uniformly sampled, so that $q^- = 1/K$, where K represents the number of classes when $K > 2$. As a result, ECE is positively correlated with the expectation of the confidence value for randomly sampled pairs $\mathbb{E}_{(v,v')}[\sigma(h_v, h_{v'})]$.

A.2 Contrast-Reg Benifits to Mitigate the Miscalibration in Graph Contrastive Learning

Figure 2a and Figure 2b investigate the impact of Contrast-Reg on the ECE value for the Pubmed dataset across epochs. In this section, we provide a comparison of the impact on ECE value when the loss converges between models with and without Contrast-Reg, across various datasets and contrastive strategies in Table 6. Table 6 demonstrates that, with Contrast-Reg, ECE values decrease for all tested datasets and contrastive losses. The reduced ECE indicates that Contrast-Reg promotes better alignment with the performance of downstream tasks while simultaneously minimizing the training loss, ensuring that minimizing the contrastive loss with Contrast-Reg leads to high-quality representations.

A.3 Explanation of Contrast-Reg's Impact on the Term $s(f)$

In this section, we provide an explanation for why Contrast-Reg leads to a decrease in the upper bound of $s(f)$. Firstly, we will prove that minimizing Eq. (7) results in a decrease in $\text{Var}(\|h_i\|)$ when $h_i^T W\mathbf{r} > c$, as stated in Theorem 2. Then, based on the assumption that models with lower $\text{Var}(\|h_i\|)$ inherently favor lower values of $\mathbb{E}[\|h_i\|]$, both $\mathbb{E}_{v_i \sim D_c}[\|h_i\|^2]$ and $\sigma_{max}(M(f,c))$ will decrease. Consequently, the upper bound of $s(f)$ decreases with the implementation of Contrast-Reg. In the subsequent proof, $h_i = f(x)$, and the notations h_i and $f(x)$ may be used interchangeably for the sake of presentation clarity.

Theorem 2. *Minimizing Eq. (7) induces the decrease in* $\text{Var}(\|h_i\|)$ *when* $h_i^T W\mathbf{r} > c$.

Proof. We minimize $\mathcal{L}_{reg}$ by gradient descent with learning rate β.

$$\frac{\partial}{\partial f(x)}\mathcal{L}_{reg} = -\sigma(-f(x)^T W\mathbf{r})W\mathbf{r} \tag{12}$$

The embedding of $f(x)$ is updated as the following after adding $\mathcal{L}_reg$:

$$f(x) \leftarrow f(x) + \beta\left(\sigma(-f(x)^T W\mathbf{r})W\mathbf{r}\right) \tag{13}$$

Eq. (13) shows that in every optimization step, $f(x)$ extends by $\beta\sigma(-f(x)^T W\mathbf{r})\left\|W\mathbf{r}\right\|$ along $\mathbf{r}_0 \coloneqq \frac{W\mathbf{r}}{\|W\mathbf{r}\|}$. If we do orthogonal decomposition for $f(x)$ along $\mathbf{r}_0$ and its unit orthogonal hyperplain $\Pi(\mathbf{r}_0)$, $f(x) = \left(f(x)^T\mathbf{r}_0\right)\mathbf{r}_0 + \left(f(x)^T\Pi(\mathbf{r}_0)\right)\Pi(\mathbf{r}_0)$. Thus we have

$$\|f(x)\| = \sqrt{(f(x)^T\mathbf{r}_0)^2 + (f(x)^T\Pi(\mathbf{r}_0))^2}. \tag{14}$$

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

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

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

Based on Lemma 1 and Eq. (14),

when $\beta\left\|W\mathbf{r}\right\|^2 \leq 1$ and $f(x)^T W\mathbf{r} > 1.5$, we have

$$\operatorname{Var}\left(\left\|(f(x))_{reg}\right\|\right) < \operatorname{Var}\left(\|f(x)\|\right). \tag{15}$$

□

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

$$\operatorname{Var}\left(\sqrt{(X + \frac{\tau}{1+e^X})^2 + c^2}\right) < \operatorname{Var}\left(\sqrt{X^2 + c^2}\right). \tag{16}$$

Proof. First, we consider

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

where $h(x)$ is strictly decreasing in $[x_0, +\infty)$ and strictly increasing in $(-\infty, x_0]$, and x_0 is the solution of $h'(x) = \frac{\mathrm{d}h(x)}{\mathrm{d}x} = 0$. 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 τ and c^2.

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

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

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

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

When $y > x \geq 1.5$,

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

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

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

□

Remark 1. $\beta\left\|W\mathbf{r}\right\|^2 \le 1$, *which is the condition of Eq. (15) , 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 2 is not a tight bound for* x_0 *in Lemma 1. Since when Eq. (7) converges,* $f(x)^T W\mathbf{r}$ *is much larger than 1.5 for almost all the samples empirically, we prove the case for* $f(x)^T w\mathbf{r}\in[1.5,+\infty)$*.*

A.4 Comprehensive Explanation of Theorem 1: Notations and Proof

To formally analyze the behavior of contrastive learning, Saunshi et al. (2019) introduce the following concepts.

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

  $$\mathcal{D}_{sim}(x,x^+)=\mathbb{E}_{c\in\rho}\left[\mathcal{D}_c(x)\mathcal{D}_c(x^+)\right], \tag{17}$$

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

  $$\mathcal{D}_{neg}(x^-)=\mathbb{E}_{c\in\rho}\left[\mathcal{D}_c(x^-)\right] \tag{18}$$

- *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 $\mathcal{T}\subseteq\mathcal{C}$ with $|\mathcal{T}|=K+1$.
  Mean classifier W^μ is naturally imposed to bridge the gap between the representation learning performance and linear separability of learn representations, as

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

- *Empirical Rademacher complexity*: Suppose $\mathcal{F}:\mathcal{X}\to[1,0]$. 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 $\vec{e}=(e_1,\cdots,e_m)^T$, with e_i are independent random variables taking values uniformly from $\{-1,+1\}$.

In addition, the theoretical framework in Saunshi et al. (2019) makes an assumption: encoder f is bounded, i.e., $\max_{x\in\mathcal{X}} \|f(x)\| \leq R^2$, $R \in \mathbb{R}$.

To prove Theorem 1, we first list some key lemmas.

Lemma 2. *For all $f \in \mathcal{F}$,*

$$\mathcal{L}^{\mu}_{sup}(f) \leq \frac{1}{1-\tau}(\mathcal{L}_{nce}(f) - 2\tau\log 2). \tag{19}$$

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.6.

Lemma 3. *With probability at least $1-\delta$ over the set $\mathcal{S}$, for all $f \in \mathcal{F}$,*

$$\mathcal{L}_{nce}(\hat{f}) \leq \mathcal{L}_{nce}(f) + Gen_M. \tag{20}$$

This bound guarantees that the chosen $\hat{f} = \arg\min_{f\in\mathcal{F}} \mathcal{L}^{\mu}_{nce}$ 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 Mohri et al. (2018) and vector-contraction inequality Maurer (2016). More details are given in Appendix A.7.

Lemma 4. $\mathcal{L}^{=}_{nce}(f) \leq 4s(f) + 2\log 2$.

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.8).

Proof to Theorem 1. Combining Lemma 2 and Lemma 3, we obtain with probability at least $1-\delta$ over the set $\mathcal{S}$, for all $f \in \mathcal{F}$,

$$\mathcal{L}^{\mu}_{sup}(\hat{f}) \leq \frac{1}{1-\tau}\left(\mathcal{L}_{nce}(f) + Gen_M\right) \tag{21}$$

Then, we decompose $\mathcal{L}_{nce} = \tau\mathcal{L}^{=}_{nce}(f) + (1-\tau)\mathcal{L}^{\neq}_{nce}(f)$, apply Lemma 4 to Eq. (21), and obtain the result of Theorem 1 ☐

A.5 Contrastive Learning with NCEloss

The contrastive loss defined by Saunshi et al. (2019) is

$$\mathcal{L}_{un} \coloneqq \mathop{\mathbb{E}}_{\substack{(x,x^+)\sim\mathcal{D}_{sim},\\ (x^-_1,\cdots,x^-_K)\sim\mathcal{D}_{neg}}} \left[\ell(\{f(x)^T(f(x^+) - f(x^-_i))\}^K_{i=1})\right],$$

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

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

A more powerful loss function, NCEloss, used in Velickovic et al. (2019); Yang et al. (2020b); Mnih & Teh (2012); Dyer (2014), can be framed as

$$\mathcal{L}_{nce} \coloneqq - \mathop{\mathbb{E}}_{\substack{(x,x^+)\sim\mathcal{D}_{sim},\\ (x^-_1,\cdots,x^-_K)\sim\mathcal{D}_{neg}}} \left[\log\sigma(f(x)^T f(x^+)) + \sum_{k=1}^{K}\log\sigma(-f(x)^T f(x^-_k))\right], \tag{22}$$

and its empirical counterpart with M samples $\left(x_i, x^+_i, x^-_{i1}, \cdots, x^-_{iK}\right)^M_{i=1}$ is given as

$$\hat{\mathcal{L}}_{nce} \coloneqq -\frac{1}{M}\sum_{i=1}^{M}\left[\log\sigma(f(x_i)^T f(x^+_i)) + \sum_{k=1}^{K}\log\sigma(-f(x_i)^T f(x^-_{ij}))\right], \tag{23}$$

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:

$$\mathcal{L}^{\mu}_{sup} := -\mathop{\mathbb{E}}_{(x,c)\sim\mathcal{D}_{\mathcal{T}}(x,c)}\left[\log\sigma(f(x)^T\mu_c)+\log\sigma(-f(x)^T\mu_{c'})\,|\,c'\neq c\right]. \tag{24}$$

A.6 Proof of Lemma 2

First, we prove that $\ell(f(x^+),\{f(x_i^-)\}) = -(\log\sigma(f(x)^T f(x^+)) + \sum_{i=1}^K \log(\sigma(f(x)^T f(x^-)))$ is convex w.r.t. $f(x^+), f(x_i^-), \cdots, f(x_K^-)$. 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 $\ell(f(x^+),\{f(x_i^-)\}) = -(\log\sigma(f(x)^T f(x^+)) + \sum_{i=1}^K \log(\sigma(f(x)^T f(x^-))))$ is convex since non-negative weighted sums preserve convexity Boyd & Vandenberghe (2014). By the definition of convexity,

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

A.7 Generalization bound

Denote

$$\begin{aligned}
\tilde{\mathcal{F}} = \big\{\tilde{f}\left(x_i, x_i^+, x_{i1}^-, \cdots, x_{iK}^-\right) = \\
\left(f(x_i), f(x_i^+), f(x_{i1}^-), \cdots, f(x_{iK}^-)\right) | f \in \mathcal{F}\big\}.
\end{aligned}$$

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

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

Denote $z_i = \left(x_i, x_i^+, x_{i1}^-, \cdots, x_{iK}^-\right)$, suppose ℓ 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

$$\begin{aligned}
\phi(\tilde{f}(z_i)) = \Bigg(\sum_{t=1}^{d} f(x_i)_t f(x_{i0}^+)_t, \sum_{t=1}^{d} f(x_i)_t f(x_{i1}^-)_t, \cdots, \\
\sum_{t=1}^{d} f(x_i)_t f(x_{iK}^-)_t\Bigg)
\end{aligned} \tag{25}$$

$$\ell(\mathbf{x}) = -\left(\log\sigma(x_0) + \sum_{i=1}^{K}\log\sigma(-x_i)\right).$$

From Eq. (25), we know that $\phi : \mathbb{R}^{(K+2)d} \to \mathbb{R}^{K+1}$.
Then we will prove that h is L-Lipschitz by proving that ϕ and ℓ are both Lipschitz continuity. First,

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

$$\frac{\partial\phi(\tilde{f}(z_i))}{f(x_{i0}^+)_t} = f(x_i)_t, \quad \frac{\partial\phi(\tilde{f}(z_i))}{f(x_{ik}^-)_t} = f(x_i)_t.$$

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 ϕ is $\sqrt{2(K+1)}R$-Lipschitz. Similarly, ℓ 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 inequalityMaurer (2016). We have

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

If we write it in Rademacher complexity manner, we have

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

Applying generalization bounds based on Rademacher complexity Mohri et al. (2018) to $q \in \mathcal{Q}$. For any $\delta > 0$, with the probability of at least $1 - \frac{\delta}{2}$,

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

Thus for any f,

$$\mathcal{L}_{nce}(f) \leq \tilde{\mathcal{L}}_{nce}(f) + \frac{4(K+1)R\mathcal{R}_{\mathcal{S}}(\mathcal{F})}{M} + 3B\sqrt{\frac{\log\frac{4}{\delta}}{2M}}. \tag{26}$$

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 $1 - \frac{\delta}{2}$,

$$\tilde{\mathcal{L}}_{nce}(f^*) \leq \mathcal{L}_{nce}(f^*) + B\sqrt{\frac{\log\frac{2}{\delta}}{2M}} \tag{27}$$

Substituting $\hat{f}$ into Eq. (26), combining $\tilde{\mathcal{L}}_{nce}(\hat{f}) \leq \mathcal{L}_{nce}(f^*)$ and applying union bound, with probability of at most δ

$$\begin{aligned}\mathcal{L}_{nce}(\hat{f}) \leq& \tilde{\mathcal{L}}_{nce}(\hat{f}) + \frac{4(K+1)R\mathcal{R}_{\mathcal{S}}(\mathcal{F})}{M} + 3B\sqrt{\frac{\log\frac{4}{\delta}}{2M}} + B\sqrt{\frac{\log\frac{2}{\delta}}{2M}} \\ &\leq \mathcal{L}_{nce}(f^*) + \frac{4(K+1)R\mathcal{R}_{\mathcal{S}}(\mathcal{F})}{M} + 4B\sqrt{\frac{\log\frac{4}{\delta}}{2M}} \\ &\leq \mathcal{L}_{nce}(f) + \frac{4(K+1)R\mathcal{R}_{\mathcal{S}}(\mathcal{F})}{M} - 4(K+1)\log(\sigma(-R^2))\sqrt{\frac{\log\frac{4}{\delta}}{2M}}\end{aligned} \tag{28}$$

fails. Thus, with probability of at least $1 - \delta$, Eq. (28) holds.

Table 7: GAT as encoder with and w/o Contrast-Reg

Algorithm	Cora	Wiki	Computers
ML	76.90±1.20	69.02±1.89	78.47±2.60
ML+Contrast-Reg	**81.56**±0.94	**69.41**±0.85	**80.41**±1.96
LC	74.30±1.10	61.55±1.89	73.27±5.98
LC+Contrast-Reg	**80.87**±0.96	**69.92**±0.80	**80.28**±2.09

A.8 Class collision loss

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

$$\begin{aligned}\mathcal{L}^{=}_{nce}(f) &= -\mathbb{E}\left[\log\sigma(f(x)^T f(x_0^+)) + \sum_{i=1}^{K}\log\sigma(-f(x)^T f(x_i^-)))\right]\\ &= \mathbb{E}\left[\log(1+e^{-f(x)^T f(x_0^+)}) + \sum_{i=1}^{K}\log(1+e^{f(x)^T f(x_i^-)})\right]\\ &\le (K+1)\mathbb{E}\left[\log(1+e^{p})\right]\\ &\le (K+1)\log 2 + (K+1)\mathbb{E}\left[p\right]\end{aligned} \tag{29}$$

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

$$\mathbb{E}[p] = \int P[p \ge x]dx = \int (1-(1-P[p_0 \ge x])^{K+1}dx. \tag{30}$$

Applying Bernoulli's inequality, we have

$$\begin{aligned}\mathbb{E}[p] &\le \int (1-(1-(K+1)P[p_0 \ge x]))dx\\ &= \int (K+1)P[p_0 \ge x]dx\\ &= (K+1)\mathbb{E}[p_0]\\ &= (K+1)\mathbb{E}[|f(x)^T f(x_0^+)|]\\ &\le (K+1)\sqrt{\mathbb{E}[(f(x)^T f(x_0^+))^2]}.\end{aligned} \tag{31}$$

Therefore,

$$\mathcal{L}^{=}_{nce}(f) \le (K+1)\log 2 + (K+1)^2 s(f) \tag{32}$$

A.9 Contrat-Reg Ablation Study on GAT Backbone

To showcase the efficacy of Contrast-Reg on graph contrastive learning performance with various backbones, we conduct additional full-batch experiments using the Graph Attention Network (GAT) Velickovic et al. (2018) as the encoder backbone. The results, presented in Table 7, demonstrate that Contrast-Reg effectively calibrates the contrastive model on the GAT backbone for node classification tasks across all three datasets. These results, in conjunction with the full-batch Graph Convolutional Network (GCN) Kipf & Welling (2017) encoder and the mini-batch GraphSAGE Hamilton et al. (2017) encoder used in Tables 1-5, highlight Contrast-Reg's capacity to effectively work with various encoder backbones, f, which is consistent with the analysis presented in Section 4.

B Experiment details

Dataset statistics The dataset statistics is shown in Table 8.

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.

Table 8: Datasets

Dataset	Node #	Edge #	Feature #	Class #
Cora Yang et al. (2016)	2,708	5,429	1,433	7
Citeseer Yang et al. (2016)	3,327	4,732	3,703	6
Pubmed Yang et al. (2016)	19,717	44,338	500	3
ogbn-arxiv Hu et al. (2020a)	169,343	1,166,243	128	40
Wiki Yang et al. (2015)	2,405	17,981	4,973	3
Computers Shchur et al. (2018)	13,381	245,778	767	10
Photo Shchur et al. (2018)	7,487	119,043	745	8
ogbn-products Hu et al. (2020a)	2,449,029	61,859,140	100	47
Reddit Hamilton et al. (2017)	232,965	114,615,892	602	41

Software Configuration: Our models, as well as the DGI, GMI and GCN baselines, were implemented in PyTorch Geometric Fey & Lenssen (2019) version 1.4.3, DGL Wang et al. (2019) 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.