Multi-Scale Contrastive Siamese Networks for Self-Supervised Graph Representation Learning

Ming Jin[1] , **Yizhen Zheng**[1] , **Yuan-Fang Li**[1] , **Chen Gong**[2] , **Chuan Zhou**[3] and **Shirui Pan**[1*]
[1]Department of Data Science and AI, Faculty of IT, Monash University, Australia
[2]School of Computer Science and Engineering, Nanjing University of Science and Technology, China
[3]Academy of Mathematics and Systems Science, Chinese Academy of Sciences, China
{ming.jin, yizhen.zheng, yuanfang.li, shirui.pan}@monash.edu, chen.gong@njust.edu.cn,
zhouchuan@amss.ac.cn

Abstract

Graph representation learning plays a vital role in processing graph-structured data. However, prior arts on graph representation learning heavily rely on labeling information. To overcome this problem, inspired by the recent success of graph contrastive learning and Siamese networks in visual representation learning, we propose a novel self-supervised approach in this paper to learn node representations by enhancing Siamese self-distillation with multi-scale contrastive learning. Specifically, we first generate two augmented views from the input graph based on local and global perspectives. Then, we employ two objectives called cross-view and cross-network contrastiveness to maximize the agreement between node representations across different views and networks. To demonstrate the effectiveness of our approach, we perform empirical experiments on five real-world datasets. Our method not only achieves new state-of-the-art results but also surpasses some semi-supervised counterparts by large margins. Code is made available at https://github.com/GRAND-Lab/MERIT

1 Introduction

Over the past few years, graph representation learning, which aims to learn low-dimensional embeddings of nodes or graphs to preserve the underlying structural and attributive information, has become a pivotal part of mining graph-structured data. The learned embeddings can then be used in different downstream tasks, such as node and graph classification, by training specific decoders on top of the learned embeddings. Although graph neural networks (GNNs) [Defferrard *et al.*, 2016; Kipf and Welling, 2017a; Veličković *et al.*, 2018; Hamilton *et al.*, 2017] have achieved significant progress in graph representation learning, most of them require a certain number of labeled nodes to train, which hinders them from being adopted in real-world applications where the labeling information is usually scarce and valuable.

To mitigate this gap, self-supervised graph representation learning approaches, especially those methods based on contrastive learning, have recently achieved promising results. Traditional unsupervised methods such as DeepWalk [Perozzi *et al.*, 2014] and Node2Vec [Grover and Leskovec, 2016] are based on random walks and the skip-gram model, forcing neighboring nodes to have similar representations. However, random walk and other matrix reconstruction-based approaches [Kipf and Welling, 2017b; Hamilton *et al.*, 2017] place a strong emphasis on the graph proximity and fail to consider other widely available relationships within or between subgraphs. Following the concept of mutual information (MI) [Oord *et al.*, 2018] and other visual representation learning advances [Tian *et al.*, 2020; He *et al.*, 2020; Chen *et al.*, 2020; Hjelm *et al.*, 2019], a series of graph contrastive learning (GCL) methods have been proposed. For example, inspired by Deep InfoMax [Hjelm *et al.*, 2019], DGI [Veličković *et al.*, 2019] proposes to maximize the mutual information between the patch- and global-level representations. Based on this, MVGRL [Hassani and Khasahmadi, 2020] introduces the concept of graph multi-view contrastive learning by discriminating the patch-global representations over two augmented views that derived from the input graph. Other approaches, such as GMI [Peng *et al.*, 2020] and GRACE [Zhu *et al.*, 2020], extend the idea of MI maximization to contrast the representation of a node with its raw information (e.g., node features) or neighbors' representations in different views.

Although the aforementioned methods have achieved significant success, they suffer all or at least partially the following limitations. Firstly, existing MI-based methods, such DGI, GMI, and MVGRL, usually require an additional MI estimator to score positive (e.g., local-global representations) and negative pairs (e.g., representations from corrupted views), which is computationally expensive and also makes the model sensitive to the choice of discriminators [Tschannen *et al.*, 2020]. Secondly, most of existing GCL methods heavily rely on a large number of negative samples to avoid collapsing to trivial solutions. In other words, negative nodes and graphs act as an indispensable regulator that needs to be deliberately selected in contrastive learning. To alleviate this problem, Grill et al. [2020] propose the Bootstrap Your Own Latent (BYOL) framework to perform unsupervised representation learning on images by leveraging the bootstrapping mechanism without using negative samples. Self-supervised representation learning

*Corresponding author

methods utilizing Siamese networks [Chen and He, 2021; Shi *et al.*, 2020] predominantly work in the visual domain but have not been extended to graphs yet.

In this paper, to alleviate the drawbacks of existing GCL methods and take advantage of bootstrapping in Siamese networks, we propose a simple yet powerful framework to learn node-level representations, which we refer to as Multi-scalE contRastive sIamese neTwork (MERIT). Our method is designed to optimize two objectives, namely cross-network and cross-view contrastiveness. Firstly, considering that existing GCL methods heavily rely on negative samples to avoid representation collapse, we propose to use a momentum-driven Siamese architecture as our backbone to maximize the similarity between node representations in different views from the online and target network, respectively. The intuition behind is that the slowly-moving target network in our framework serves as a stable "mean teacher" to encode historical observations, which guides the online network to learn to explore richer and better representations without relying on extra negatives to avoid collapse [Grill *et al.*, 2020]. However, merely optimizing this objective ignores the rich underlying graph topological information. To partially alleviate this problem, we inject additional negative samples to push disparate nodes away in different views across two networks, as shown in Figure 2(a). Secondly, different from the work in the visual domain where the similarity measurements are typically defined on the image level, we propose to further utilize node connectivity and introduce a multi-scale contrastive learning within and across views in the online network, i.e., Figure 2(b), to regularize the training of the aforementioned bootstrapping objective in our method. Experimental results on a variety of datasets demonstrate the superb performance of our design.

Our contribution is summarized as follow:

- We propose a novel framework to learn node representation by taking advantage of bootstrapping in Siamese network and multi-scale graph contrastive learning. To the best of our knowledge, we are the first to use the Siamese networks on node representation learning.
- We propose two types of contrastive objectives for self-supervised node representation learning based on Siamese networks. They regularize each other and are capable of providing a more effective graph encoder.
- We conduct extensive experiments on various real-world datasets and validate the effectiveness of the proposed method over state-of-the-art methods in self-supervised graph representation learning.

2 Related Work

Siamese network is a neural architecture that contains two or more identical structures (e.g., online and target encoder in Figure 1) to make multi-class prediction or entity comparison [Bromley *et al.*, 1993]. Traditionally, it has been used on supervised tasks such as signature verification [Bromley *et al.*, 1993] and face matching [Taigman *et al.*, 2014]. Recently, Grill et al. [2020] employed this architecture on self-supervised visual representation learning and achieved significant improvements over existing arts without using negative samples. To fully understand the underlying mechanism of BYOL, SimSiam [Chen and He, 2021] and RAFT [Shi *et al.*, 2020] verify that the extra predictor in the online network and the stop-gradient mechanism in the target network are keys to prevent the collapse without the help of negative samples.

Unsupervised graph representation learning approaches are traditional based on random walks [Perozzi *et al.*, 2014; Grover and Leskovec, 2016] and adjacency matrix reconstruction [Kipf and Welling, 2017b]. These methods heavily rely on node proximity but are less scalable and error-prone without extracting other widely available self-supervision signals in graphs. Recently, unsupervised GNN-based methods, such as GraphSAGE [Hamilton *et al.*, 2017], have achieved considerable progress but still limited in performance. Contrastive methods, on the other hand, alleviate the aforementioned problems and achieve state-of-the-art performance. For example, DGI [Veličković *et al.*, 2019] and InfoGraph [Sun *et al.*, 2019] employ the idea of Deep InfoMax [Hjelm *et al.*, 2019] and consider both patch and global information during the discrimination. MVGRL [Hassani and Khasahmadi, 2020] introduces augmented views to graph contrastive learning and optimizes the DGI-like objectives. GMI [Peng *et al.*, 2020] proposes considering both graph proximity and feature space similarity by maximizing the mutual information between the representation and feature space with learnable weights. Other approaches, such as CG3 [Wan *et al.*, 2021] and GRACE [Zhu *et al.*, 2020], further extend the idea of graph MI maximization and conduct the discrimination on different scales.

3 Proposed Method

Problem definition. Given a graph $\mathcal{G} = (X, A)$, where $X \in \mathbb{R}^{N \times D}$ denotes the node feature matrix, and $A \in \mathbb{R}^{N \times N}$ indicates the adjacency matrix where each entry A_{ij} is the linkage relation between nodes i and j. In this paper, we aim to learn a graph encoder $g_\theta : \mathbb{R}^{N \times D} \times \mathbb{R}^{N \times N} \to \mathbb{R}^{N \times D'}$ such that $D' \ll D$, without relying on the labeling information. The resulted representations $H = g_\theta(X, A) = \{h_1, h_2 ..., h_N\}$ can then be directly used in downstream tasks, such as node classification.

Overall framework. We propose a novel algorithm, namely MERIT, to learn node representations by taking advantage of both bootstrapping and multi-scale graph contrastive learning. As illustrated in Figure 1, our model mainly consists of three components: Graph augmentations, cross-network contrastive learning, and cross-view contrastive learning. To train our model, we first generate two augmented graph views, denoted as $\tilde{\mathcal{G}}_1$ and $\tilde{\mathcal{G}}_2$. After this, by processing these two views via the *online network* and the *target network*, we construct different graph contrastive paths on multiple scales in the latent space, as shown on the right-most part in Figure 1. In the following sections, we illustrate the aforementioned crucial components.

3.1 Graph Augmentations

Augmentation is a key component in self-supervised visual representation learning. However, image augmentations such

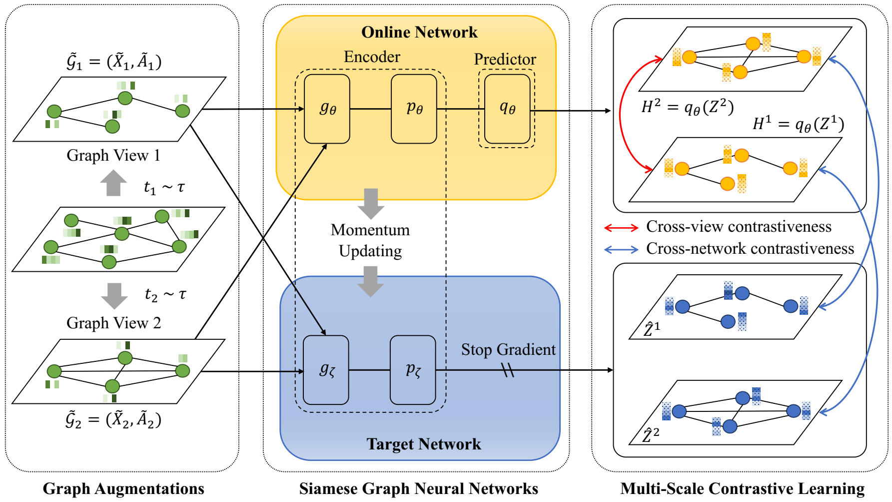


Figure 1: The overall framework of MERIT. Through graph augmentations, we construct two graph views, based on which an online network and a target network are employed to generate node representations for each view. A multi-scale contrastive learning scheme, which utilizes both cross-network and cross-view contrastive modules, is deployed to learn effective node embeddings. g_θ and g_ζ denotes a GNN-based graph encoder. p_θ, p_ζ, and q_θ are two-layer MLP with the batch normalization. $t_1 \sim \tau$ and $t_2 \sim \tau$ are two different graph augmentations

as cropping and rotating cannot be directly applied to graphs due to the huge disparity of these two modalities. Therefore, to facilitate contrastive learning on graphs, we propose four augmentation methods, as shown below, to augment the graph topological and attributive information.

Graph Diffusion (GD). We transform a graph via diffusion to generate a congruent view. The effectiveness of this method may be attributed to the extra global information provided by the diffused view. This process is fomulated as:

$$S = \sum_{k=0}^{\infty} \theta_k T^k \in \mathbb{R}^{N \times N}, \quad (1)$$

where θ is a parameter to control the distribution of local and global signals, $T \in \mathbb{R}^{N \times N}$ is the transition matrix to transfer the adjacency matrix [Klicpera *et al.*, 2019]. In this paper, we adopt the Personalized PageRank (PPR) kernel to power the graph diffusion. Formally, given the adjacency matrix A, the identity matrix I, and the degree matrix D, Equation 1 can be reformulated as:

$$S = \alpha \left(I - (1-\alpha) D^{-1/2} A D^{-1/2} \right)^{-1}, \quad (2)$$

where α is a tunable parameter for the random walk teleport probability [Hassani and Khasahmadi, 2020].

Edge Modification (EM). Instead of merely dropping edges in the adjacency matrix, we also add the same number of dropped edges [You *et al.*, 2020]. In such a way, we can maintain the original graph's property, while complicate the augmented view with the additional edges. Specifically, given the adjacency matrix A and the modification ratio P, we randomly drop $P/2$ portion of existing edges in the original graph and then randomly add the same portion of new edges to the graph. Both our edge dropping and adding process follow an i.i.d. uniform distribution.

Subsampling (SS). Similar to the image cropping, we randomly select a node index in the adjacency matrix as the splitting point, and then use it to crop the original graph to create a fixed-sized subgraph as the augmented graph view. An advantage of SS is enabling the batch processing to handle large graphs whose size may exceed the capability of the GPU memory.

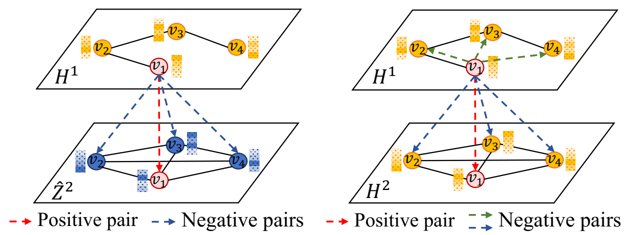
Node Feature Masking (NFM). Different from [You *et al.*, 2020], given the feature matrix X and an augmentation ratio P, we randomly select P fraction of node feature dimensions in X and then mask them with zeros.

In this paper, we apply SS, EM, and NFM to the first view, and use SS + GD + NFM for the second congruent view. By doing so, our model can encode both the local and global information through the contrastive learning.

3.2 Cross-Network Contrastive Learning

In MERIT, we introduce a Siamese architecture, which consists of two identical encoders (i.e., g_θ, p_θ, g_ζ, and p_ζ) with an extra predictor q_θ on top of the online encoder, as shown in Figure 1. We first take node representations from one view in the online network as the anchor, then we maximize the cosine similarity to the corresponding representations from another view in the target network to form the basic bootstrapping contrastiveness.

This contrastive learning process is illustrated in Figure 2(a), where $H^1 = q_\theta(Z^1)$ and $\hat{Z}^2$ denotes the representation of $\tilde{\mathcal{G}}_1$ and $\tilde{\mathcal{G}}_2$ from two different networks. Specifically, we use $Z^1 = p_\theta\left(g_\theta(\tilde{X}_1, \tilde{A}_1)\right)$ and $\hat{Z}^2 = p_\zeta\left(g_\zeta(\tilde{X}_2, \tilde{A}_2)\right)$ to denote the output node embeddings of our online and target encoder for the view 1 and 2. The red dash line between two v_1 nodes represents a positive pair $(h^1_{v_1}, \hat{z}^2_{v_1})^+$ constructed based on v_1. The intuition behind is to pull closer the representations of the same node from different views across



(a) Cross-network contrastiveness. (b) Cross-view contrastiveness.

Figure 2: Cross-network contrastive learning is based on pairs from two different representations in the online and target network. Differently, cross-view contrastiveness discriminates the pair representations from two views in the same (i.e., online) network.

two networks to distill the knowledge from historical observations, as well as stabilizing the online encoder training. To facilitate this, our target network does not directly receive the gradient during the training. Instead, we update its parameters by leveraging the momentum updating mechanism:

$$\zeta^t = m \cdot \zeta^{t-1} + (1-m) \cdot \theta^t, \tag{3}$$

where m, ζ, and θ are momentum, target network parameters, and online network parameters, respectively.

To further explore the rich contrastive relations between node representations within H^1 and $\hat{Z}_2$, we construct extra negative samples to regularize the basic bootstrapping loss, which are the blue dash lines between the red anchor node and blue nodes in Figure 2(a), i.e., $(h^1_{v_1}, \hat{z}^2_{v_j})^-$, where we aim to push away from each other. Thus, the aforementioned processes can be formulated with the following loss functions:

$$\mathcal{L}^1_{cn}(v_i) = -\log \frac{\exp(\text{sim}(h^1_{v_i}, \hat{z}^2_{v_i}))}{\sum_{j=1}^{N} \exp(\text{sim}(h^1_{v_i}, \hat{z}^2_{v_j}))}, \tag{4}$$

$$\mathcal{L}^2_{cn}(v_i) = -\log \frac{\exp(\text{sim}(h^2_{v_i}, \hat{z}^1_{v_i}))}{\sum_{j=1}^{N} \exp(\text{sim}(h^2_{v_i}, \hat{z}^1_{v_j}))}. \tag{5}$$

In above formulas, $\mathcal{L}^1_{cn}$ and $\mathcal{L}^2_{cn}$ are two symmetric losses, which represent the multi-scale cross-network contrastiveness on different views. Besides, $h^1_{v_i} \in H^1$, $h^2_{v_i} \in H^2$, $\hat{z}^1_{v_i} \in \hat{Z}^1$, $\hat{z}^2_{v_i} \in \hat{Z}^2$, and $\text{sim}(\cdot)$ denotes the cosine similarity.

Finally, by combining above two losses, we have our final cross-network contrastive objective function defined below:

$$\mathcal{L}_{cn} = \frac{1}{2N} \sum_{i=1}^{N} \left(\mathcal{L}^1_{cn}(v_i) + \mathcal{L}^2_{cn}(v_i) \right). \tag{6}$$

3.3 Cross-View Contrastive Learning

Apart from the contrastive relations across two networks, the ties between two views within the online network have not been considered yet, which acts as a strong regularization to enhance the learning ability of our method. We do not have to include such contrastive relations within the target network because it will not directly receive the gradient, and our end goal is to train g_θ within the online encoder. Figure 2(b) illustrates our cross-view contrastiveness design, which consists of two discrimination schemes from two perspectives, namely the intra- and inter-view contrastiveness. Similar to but different from GRACE [Zhu *et al.*, 2020], we apply such contrastive learning on the top of differently-augmented views, where we consider not only the local structural and attributive augmentations (e.g., edge modification and feature masking) but also the global topological information injected via the graph diffusion.

We start from the inter-view contrastiveness, which pulls closer the representations of the same nodes in two augmented views while pushing other nodes away, as depicted by the red and blue dash lines in Figure 2(b). In this case, we define our positive and negative pairs as $(h^1_{v_1}, h^2_{v_1})^+$ and $(h^1_{v_1}, h^2_{v_j})^-$. Similar to the objective functions used in the previous section, the inter-view contrastive loss $\mathcal{L}_{inter}$ for view 1 can be formulated as:

$$\mathcal{L}^1_{inter}(v_i) = -\log \frac{\exp(\text{sim}(h^1_{v_i}, h^2_{v_i}))}{\sum_{j=1}^{N} \exp(\text{sim}(h^1_{v_i}, h^2_{v_j}))}. \tag{7}$$

We can obtain $\mathcal{L}^2_{inter}(v_i)$ in similar way for view 2. On the other hand, as the red and green dash lines shown in Figure 2(b), the intra-view contrastiveness regards all nodes except the anchor node (i.e., v_1) as negatives within a particular view, denoted as $(h^1_{v_1}, h^1_{v_j})^-$. Thus, the intra-view contrastive loss $\mathcal{L}_{intra}$ for view 1 can be constructed below, which shares the same positive pairs with our inter-view contrastive losses:

$$\begin{aligned} \mathcal{L}^1_{intra}(v_i) &= -\log \frac{\exp(\text{sim}(h^1_{v_i}, h^2_{v_i}))}{\exp(\text{sim}(h^1_{v_i}, h^2_{v_i})) + \Phi}, \\ \Phi &= \sum_{j=1}^{N} \mathbb{1}_{i \neq j} \exp(\text{sim}(h^1_{v_i}, h^1_{v_j})), \end{aligned} \tag{8}$$

where Φ denotes the accumulated similarity of negative pairs in the intra-view contrastive learning. Similarly, we can calculate $\mathcal{L}^2_{intra}(v_i)$ for view 2.

By combining the inter- and intra-view contrastiveness of both views, we have our cross-view contrastive objective function $\mathcal{L}_{cv}$ formulated below:

$$\mathcal{L}_{cv} = \frac{1}{2N} \sum_{i=1}^{N} \left(\mathcal{L}^1_{cv}(v_i) + \mathcal{L}^2_{cv}(v_i) \right), \tag{9}$$

where $\mathcal{L}^1_{cv}(v_i)$ and $\mathcal{L}^2_{cv}(v_i)$ are two symmetric losses that represent the multi-scale cross-view contrasitveness on the two views:

$$\mathcal{L}^k_{cv}(v_i) = \mathcal{L}^k_{intra}(v_i) + \mathcal{L}^k_{inter}(v_i), \quad k \in \{1, 2\}. \tag{10}$$

3.4 Model Training

To train our model end-to-end and learn node representations for downstream tasks, we jointly leverage both the cross-view and cross-network contrastive loss. Specifically, the overall objective function is defined as:

$$\mathcal{L} = \beta \mathcal{L}_{cv} + (1-\beta) \mathcal{L}_{cn}, \tag{11}$$

where we aim to minimize $\mathcal{L}$ during the optimization, and β is a balance factor. During the inference, we aggregate the representations generated by the online graph encoder g_θ, taking both the graph adjacency and diffusion matrices as inputs: $\tilde{H} = H^1 + H^2 \in \mathbb{R}^{N \times D'}$, for downstream tasks.

Dataset	Nodes	Edges	Features	Classes
Cora	2,708	5,429	1,433	7
CiteSeer	3,327	4,732	3,703	6
PubMed	19,717	44,338	500	3
Amazon Photo	7,650	119,081	745	8
Coauthor CS	18,333	81,894	6,805	15

Table 1: The statistics of the datasets.

4 Experiment

To evaluate the effectiveness of MERIT on self-supervised node representation learning, we conduct extensive experiments on five widely used benchmark datasets, including Cora, CiteSeer, PubMed, Amazon Photo [Shchur *et al.*, 2018], and Coauthor CS [Shchur *et al.*, 2018]. The dataset statistics are summarized in Table 1.

4.1 Experimental Settings

For simplicity, we adopt a 1-layer GCN [Kipf and Welling, 2017a] as our backbone graph encoders (i.e., g_θ and g_ζ). For model tuning, we perform the grid search on primary hyper-parameters over certain ranges. The latent dimension of graph encoders, projectors, and the predictor is fixed to 512. We tune momentum m and augmentation ratio P between 0 and 1. To balance the effect of two contrastive schemes, we tune β within $\{0.2, 0.4, 0.6, 0.8\}$.

To evaluate the trained graph encoder, we adopt a linear evaluation protocol by training a separate logistic regression classifier on top of the learned node representations. For Cora, Citeseer and PubMed, we follow the same data splits as in [Yang *et al.*, 2016]. For Amazon Photo and Coauthor CS, we include 30 randomly-selected nodes per class to construct the training and validation set, while using the remaining nodes as the testing set. We repeat the experiments in Table 2 and 3 ten times and report the average accuracy with the standard deviation.

4.2 Classification Results

We choose node classification as our downstream task and compare MERIT with five supervised methods and four state-of-the-arts graph contrastive learning models. For supervised baselines, we select LP [Zhu *et al.*, 2003], ChebConv [Defferrard *et al.*, 2016], GCN [Kipf and Welling, 2017a], GAT [Veličković *et al.*, 2018], and SGC [Wu *et al.*, 2019]. DGI [Veličković *et al.*, 2019], MVGRL [Hassani and Khasahmadi, 2020], GMI [Peng *et al.*, 2020], and GRACE [Zhu *et al.*, 2020] are chosen as the self-supervised competitors. We report the overall classification results in Table 2 and highlight the best performance in bold.

We can observe from the table that MERIT achieves the best classification accuracy on all five datasets, surpassing not only the self-supervised but also supervised methods (except a draw with GMI on PubMed). This result can be attributed to two key components in our framework: (1). Different from the compared GCL methods, we introduce a more expressive Siamese architecture to help the graph encoder distill the knowledge from historical representations and alleviate the reliance on negative samples. (2). To further realise the potential of our model, we introduce multiple contrastive routes within and across different views and networks, which not only provide a stronger regularization to our bootstrapping objective but also enrich the self-supervision signals during the optimization.

By comparing the best performances of selected supervised and self-supervised baselines, we observe that contrastive learning-based models have achieved similar or even better classification accuracy in more than one datasets, which demonstrates the effectiveness of mining rich multi-scale contrastive relations in graphs. However, there still exist performance gaps between state-of-the-art graph contrastive learning and supervised methods in Cora and Coauthor CS.

4.3 Parameter Sensitivity Study

Balance factor β and momentum m. We study two important hyper-parameters, the balance factor β in our final objective function (i.e., Equation 11) and the momentum term m in Equation 3. In Figure 3, for a fixed momentum value, we observe that a β value between 0.4 and 0.6 typically produces the best accuracies, which confirms our conjecture that the two proposed contrastive losses can regularize each other and achieve better results than only optimizing one of them (i.e., $\beta = 0$ or $\beta = 1$), where we give a detailed analyze in our ablation study. On the other hand, for a certain β value, we find that $m = 1$ usually gives a poor performance, comparing with other values between 0 and 0.999 as recommended in BYOL [Grill *et al.*, 2020] and SimSiam [Chen and He, 2021]. We conjecture that making the parameters stale in the target network may hinder the process of knowledge distillation and thus disturb the model optimization. Another interesting finding here is that our model even performs better when $m = 0$. Our hypothesis is that the real effective components in bootstrapping are the predictor and stop gradient in our framework, which we leave to research in further work.

Effect of augmentation. Apart from m and β, augmentation plays a critical role in contrastive learning. Comparing the red and blue lines in Figure 4(a), we observe that jointly considering the structure and attributive augmentations gives the best performance. When applying the edge modification only, we surprisingly find that a higher modification ratio gives a better performance. We conjecture that this is caused by the nature of contrastive learning, which requires a more

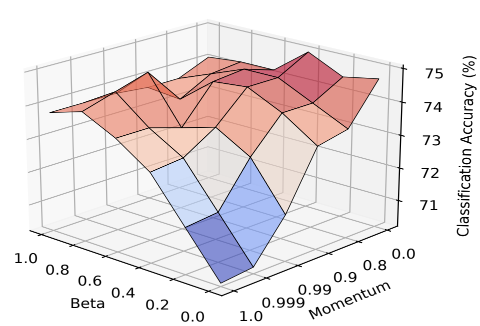
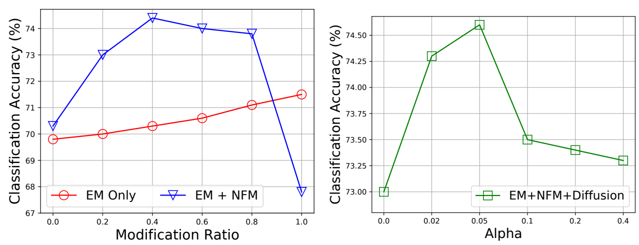


Figure 3: Classification accuracies of MERIT on CiteSeer with different β and m. A warmer color denotes a higher accuracy.

Information Used	Method	Cora	CiteSeer	PubMed	Amazon Photo	Coauthor CS
A, Y	LP	68.0	45.3	63.0	67.8±0.0	74.3 ±0.0
X, A, Y	Chebyshev	81.2	69.8	74.4	74.3±0.0	91.5 ±0.0
X, A, Y	GCN	81.5	70.3	79.0	87.3±1.0	91.8 ±0.1
X, A, Y	GAT	83.0 ±0.7	72.5 ±0.7	79.0 ±0.3	86.2 ±1.5	90.5 ±0.7
X, A, Y	SGC	81.0 ±0.0	71.9 ±0.1	78.9 ±0.0	86.4 ±0.0	91.0 ±0.0
X, A	DGI	81.7 ±0.6	71.5 ±0.7	77.3 ±0.6	83.1 ±0.5	90.0 ±0.3
X, A	GMI	82.7 ±0.2	73.0 ±0.3	**80.1 ±0.2**	85.1 ±0.1	91.0 ±0.0
X, A	MVGRL	82.9 ±0.7	72.6 ±0.7	79.4 ±0.3	87.3 ±0.3	91.3 ±0.1
X, A	GRACE	80.0 ±0.4	71.7 ±0.6	79.5 ±1.1	81.8 ±1.0	90.1 ±0.8
X, A	MERIT	**83.1 ±0.6**	**74.0 ±0.7**	**80.1 ±0.4**	**87.4 ±0.2**	**92.4 ±0.4**

Table 2: Classification accuracies on five benchmark datasets. **X**, **A**, and **Y** indicate the node feature, adjacency matrix, and label information exploited by each algorithm, respectively. Some results without standard deviations are directly taken from [Hassani and Khasahmadi, 2020].



(a) Study on structural and attributive graph augmentations. (b) Effect of graph diffusion on performance.

Figure 4: Classification accuracies versus graph augmentation in varying types and degrees.

Method	**CiteSeer**	**Amazon Photo**
MERIT	74.0 ±0.7	87.4 ±0.2
MERIT w/o cross-network	73.8 ±0.4	87.0 ±0.1
MERIT w/o cross-view	73.6 ±0.4	87.1 ±0.3

Table 3: Ablation study on CiteSeer and Amazon Photo

challenging pretext task to achieve competitive performance. When considering multi-scale augmentations, e.g., combining edge and node feature modification, overly increasing the modification ratio may distort the underlying topological and attributive information, thus leading to significant performance degrade. The effectiveness of graph diffusion can be observed in Figure 4(b), where $\alpha = 0.05$ gives the best performance in our experiments. When removing this module (i.e., $\alpha = 0$), the performance decreases dramatically, which confirms our hypothesis that injecting global information further boosts the expressive ability of our model.

4.4 Ablation Study

To validate the effectiveness of the two contrastive components, we conduct experiments on Citesser and Amazon Photo for two MERIT variants, each of which has one of the key components removed. The result is presented in Table 3. Here we use MERIT w/o cross-network and MERIT w/o cross-view to denote the ablated model with cross-network loss $\mathcal{L}_{cn}$ or cross-view loss $\mathcal{L}_{cv}$ being masked.

(a) GCN (b) DGI (c) MERIT

Figure 5: t-SNE embeddings of nodes in the CiteSeer dataset.

From Table 3, we can find that our model performance would degrade without one of the key components on the two datasets, which demonstrates the effectiveness of our two contrastive schemes. Specifically, our proposed model can boost MERIT w/o cross-view with 0.4% and 0.3% improvement, and MERIT w/o cross-network with 0.2% and 0.4% improvement for CiteSeer and Amazon Photo, respectively. This improvement can be attributed to our comprehensive multi-scale contrastive learning scheme, which takes the advantage of both single- and multiple-network contrastiveness.

Visualisation. To show the superiority of our model, we visualize the node embeddings of CiteSeer calculated by GCN, DGI, and MERIT via the t-SNE algorithm, in which node colors denote different classes. In Figure 5, MERIT's 2D projection presents a clearer separation, which indicates that our approach benefits the graph encoder to extract more expressive node representations for downstream tasks.

5 Conclusion

In this paper, we present a novel approach towards self-supervised graph representation learning. By leveraging the backbone Siamese GNNs, we design a cross-network contrastiveness to distill the knowledge from historical representations to guide and stabilize the training of online graph encoder. To further enrich the self-supervision signals, we introduce another cross-view contrastive objective on multiple scales to regularize the bootstrapping scheme in the cross-network contrastiveness. Experimental results demonstrate the superiority and the effectiveness of our method.

References

[Bromley *et al.*, 1993] J. Bromley, James W. Bentz, L. Bottou, I. Guyon, Y. LeCun, C. Moore, Eduard Säckinger, and R. Shah. Signature verification using a "siamese" time delay neural network. In *IJPRAI*, 1993.

[Chen and He, 2021] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR*, 2021.

[Chen *et al.*, 2020] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

[Defferrard *et al.*, 2016] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *NeurIPS*, 2016.

[Grill *et al.*, 2020] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent - a new approach to self-supervised learning. In *NeurIPS*, 2020.

[Grover and Leskovec, 2016] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *KDD*, 2016.

[Hamilton *et al.*, 2017] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NeurIPS*, 2017.

[Hassani and Khasahmadi, 2020] Kaveh Hassani and Amir Hosein Khasahmadi. Contrastive multi-view representation learning on graphs. In *ICML*, 2020.

[He *et al.*, 2020] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

[Hjelm *et al.*, 2019] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *ICLR*, 2019.

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

[Kipf and Welling, 2017b] Thomas N Kipf and Max Welling. Variational graph auto-encoders. In *ICLR*, 2017.

[Klicpera *et al.*, 2019] Johannes Klicpera, Stefan Weißenberger, and Stephan Günnemann. Diffusion improves graph learning. In *NeurIPS*, 2019.

[Oord *et al.*, 2018] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[Peng *et al.*, 2020] 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*, 2020.

[Perozzi *et al.*, 2014] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *KDD*, 2014.

[Shchur *et al.*, 2018] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation. In *NeurIPS Relational Representation Learning Workshop*, 2018.

[Shi *et al.*, 2020] Haizhou Shi, Dongliang Luo, Siliang Tang, Jian Wang, and Yueting Zhuang. Run away from your teacher: Understanding byol by a novel self-supervised approach. *arXiv preprint arXiv:2011.10944*, 2020.

[Sun *et al.*, 2019] Fan-Yun Sun, Jordan Hoffman, Vikas Verma, and Jian Tang. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. In *ICLR*, 2019.

[Taigman *et al.*, 2014] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*, 2014.

[Tian *et al.*, 2020] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *ECCV*, 2020.

[Tschannen *et al.*, 2020] Michael Tschannen, Josip Djolonga, Paul K Rubenstein, Sylvain Gelly, and Mario Lucic. On mutual information maximization for representation learning. In *ICLR*, 2020.

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

[Veličković *et al.*, 2019] Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. In *ICLR*, 2019.

[Wan *et al.*, 2021] Sheng Wan, Shirui Pan, Jian Yang, and Chen Gong. Contrastive and generative graph convolutional networks for graph-based semi-supervised learning. In *AAAI*, 2021.

[Wu *et al.*, 2019] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In *ICML*, 2019.

[Yang *et al.*, 2016] Zhilin Yang, William Cohen, and Ruslan Salakhudinov. Revisiting semi-supervised learning with graph embeddings. In *ICML*, 2016.

[You *et al.*, 2020] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. In *NeurIPS*, 2020.

[Zhu *et al.*, 2003] Xiaojin Zhu, Zoubin Ghahramani, and John D Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, 2003.

[Zhu *et al.*, 2020] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Deep graph contrastive representation learning. In *ICML Workshop on Graph Representation Learning and Beyond*, 2020.