Mixed Graph Contrastive Network for Semi-supervised Node Classification

XIHONG YANG, National University of Defense Technology, ChangSha, China
YIQI WANG, National University of Defense Technology, ChangSha, China
YUE LIU, National University of Defense Technology, ChangSha, China
YI WEN, National University of Defense Technology, ChangSha, China
LINGYUAN MENG, National University of Defense Technology, ChangSha, China
SIHANG ZHOU, National University of Defense Technology, ChangSha, China
XINWANG LIU and EN ZHU, National University of Defense Technology, ChangSha, China

Graph Neural Networks (GNNs) have achieved promising performance in semi-supervised node classification in recent years. However, the problem of insufficient supervision, together with representation collapse, largely limits the performance of the GNNs in this field. To alleviate the collapse of node representations in semi-supervised scenario, we propose a novel graph contrastive learning method, termed Mixed Graph Contrastive Network (MGCN). In our method, we improve the discriminative capability of the latent embeddings by an interpolation-based augmentation strategy and a correlation reduction mechanism. Specifically, we first conduct the interpolation-based augmentation in the latent space and then force the prediction model to change linearly between samples. Second, we enable the learned network to tell apart samples across two interpolation-perturbed views through forcing the correlation matrix across views to approximate an identity matrix. By combining the two settings, we extract rich supervision information from both the abundant unlabeled nodes and the rare yet valuable labeled nodes for discriminative representation learning. Extensive experimental results on six datasets demonstrate the effectiveness and the generality of MGCN compared to the existing state-of-the-art methods. The code of MGCN is available at https://github.com/xihongyang1999/MGCN on Github.

CCS Concepts: • Computing methodologies → Semi-supervised learning settings;

Additional Key Words and Phrases: Semi-supervised classification; contrastive learning; graph neural network

ACM Reference Format:
Xihong Yang, Yiqi Wang, Yue Liu, Yi Wen, Lingyuan Meng, Sihang Zhou, Xinwang Liu, and En Zhu. 2024. Mixed Graph Contrastive Network for Semi-supervised Node Classification. ACM Trans. Knowl. Discov. Data. 18, 7, Article 162 (June 2024), 19 pages. https://doi.org/10.1145/3641549

This work was supported by the National Key R&D Program of China (no. 2022ZD0209103), the National Natural Science Foundation of China (no. 62325604, 62276271), the Postgraduate Scientific Research Innovation Project in Hunan Province (No. CX20220076).

Authors’ addresses: X. Yang, Y. Wang, Y. Liu, Y. Wen, L. Meng, S. Zhou, X. Liu (Corresponding author), and E. Zhu (Corresponding author), National University of Defense Technology, Deya road No. 109, ChangSha, Hunan, China, 410073; e-mails: yangxihong@nudt.edu.cn, xinwangliu@nudt.edu.cn.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM 1556-4681/2024/06-ART162
https://doi.org/10.1145/3641549
1 INTRODUCTION

In recent years, machine learning has developed rapidly and achieved remarkable performance in many fields, such as image classification [13, 14, 44–46], recommendation [4–6, 74, 85], and clustering [24, 25, 55–57, 70, 81]. With the strong representation learning capacity, graph learning methods [32–34, 84] have become a hot research spot in many fields, including the graph clustering [23, 26, 29, 31, 72, 73], collaborative filtering [79, 80], molecular graph [66, 67, 77, 78], and so on. Semi-supervised node classification, which aims to classify nodes in the graph with limited labels, is a crucial yet challenging graph learning task. Thanks to the powerful feature extraction capability, Graph Convolutional Network (GCN) [20] has recently achieved promising performance in this scenario. As a result, it has attracted considerable attention in this field, and many methods [8, 21, 50, 68] have been proposed.

Although preferable performances have been achieved by the existing algorithms, in the semi-supervised node classification task, insufficient supervision has largely aggravated the problem of representation collapse in graph learning, leading to indiscriminate representation across classes. To solve the problem, a commonly used strategy is to path the supervision information from the labeled data to the unlabeled data according to the linkages within the adjacent matrix as guidance for network training [11, 20, 50, 59–61, 68]. Moreover, in MixupForGraph [62], a graph mixup operation is designed to enhance the robustness and discriminative capability of the aggregated sample embedding over the labeled samples. Since the embedding of the labeled samples has integrated information of both the labeled sample and its unlabeled neighbors while pushing the predictions to their corresponding ground truth, the information of the unlabeled samples is also integrated for network training in a form of implicit regularization.

Though valuable information is introduced, the performance of these methods could be significantly influenced by the inaccurate connections within the data. Recently, to alleviate the adverse influence of the inaccurate connections, MVGRL [12] introduces contrastive learning as an auxiliary task for discriminative information exploitation. In this method, the authors design an InfoMax loss to maximize the cross-view mutual information between the node and the global summary of the graph.

Although large improvement has been made, the current data augmentation and loss function setting of MVGRL fails to exploit abundant intuitive information within the unlabeled data, thus limiting its classification performance. This phenomenon can be witnessed in the cosine similarity matrix of latent representation illustration in Figure 1. As we can see, although the categorical information is revealed by the learned representations to different extent, more discriminative information is needed for further performance enhancement.

To solve this issue, we propose a novel graph contrastive semi-supervised learning method termed Mixed Graph Contrastive Network (MGCN), which improves the discriminative capability of node embedding by conducting the interpolation-based augmentation and improving the cross-view consistency of the latent representation among samples. To be specific, we first adopt the interpolation-based strategy to conduct data augmentation in the latent space and then force the prediction model to change linearly between samples as done in the field of image recognition [52]. After that, by forcing the correlation matrix across two interpolation-perturbed views to approximate an identical matrix, we guide our network to be able to recognize whether two perturbed samples are the same samples or not. In this manner, the sample representations would be more discriminative, thus alleviating the collapsed representations. This could be clearly seen in Figure 1(d) that the similarity matrix generated by our method can obviously reveal the hidden distribution structure better than the compared methods. The key contributions of this article are listed as follows:
We observe the representation collapse problem under the semi-supervised scene and propose a novel graph contrastive learning method to solve this issue.

An interpolation-based augmentation strategy and a correlation reduction mechanism are designed to improve discriminative capability of representations.

Extensive experimental results on six datasets demonstrate the superiority of our method against the compared state-of-the-art method. The ablation study and module transferring experiments demonstrate the effectiveness and the generality of our proposed modules.

2 RELATED WORK

2.1 Semi-supervised Node Classification

Semi-supervised node classification [15–17, 58] aims to classify nodes in the graph with few human annotations. Recently, Graph Neural Networks (GNNS) have achieved promising performance for their strong representation capability of graph-structured data. The pioneer GCN-Chey [9] generalizes CNN [22] to graphs in the spectral domain by proposing the Chebyshev polynomials graph filter. Following GCN-Chey, GCN [20] reveals the underlying graph structure by feature transformation and aggregation operations in the spatial domain. After that, GraphSage [11] generates embeddings by sampling and aggregating features from the node neighborhoods. GAT [50] proposed graph attention networks on graph-structured data to improve the performance. JK-Net [68] flexibly leverages different neighborhood ranges to enable better structure-aware representation. In addition, SGC [63] simplifies GCN by removing feature transformation between consecutive layers. Furthermore, Geom-GCN [39] proposes a geometric aggregation scheme to overcome the issue of neighborhood node structural information loss. Different from them, PPNP/APPNP [21] separates the feature transformation from aggregation operation and enhances the aggregation operation with PageRank [37]. More recently, following PPNP/APPNP, GPRGNN [8] jointly optimizes sample feature and topological information by learning the aggregation weights adaptively.

In our proposed method, we adopt GPRGNN [8] as our backbone and further improve its discriminative capability by conducting the interpolation-based augmentation and improving the cross-view consistency of the latent representation.

2.2 Representation Collapse

Contrastive learning methods [7, 10, 27, 30] have achieved promising performance on images in recent years. Motivated by their success, contrastive learning strategies have been increasingly adopted to the graph data [12, 28, 51, 71, 82].
The pioneer DGI [51] is proposed to learn node embedding by maximizing the mutual information between the local and global fields of the graph. GMI [40] and HDMI [18] improve DGI by regarding edges and node attributes, respectively, to alleviate collapse representation. Besides, MVGRL [12] and InfoGraph [42] demonstrate the effectiveness of maximizing the mutual information to learn graph-level representations in the graph classification task. Subsequently, GraphCL [75] and GRACE [82] first generate two augmented views and then learn node embeddings by pulling together the same node in two augmented views while pushing away different nodes. However, representation collapse is a common problem that, without the adequate guidance of human annotations, the model tends to embed all samples to the same representation.

To alleviate representation collapse, BGRL [47] is proposed to learn node embeddings by two separate GCN encoders. Specifically, the online encoder is trained to pull together the same node from two views while the target encoder is updated by an exponential moving average of online encoder. More recently, G-BT [3] is proposed to avoid representation collapse by reducing the redundancy of features. MGCN implicitly achieves the redundancy-reduction principle through an interpolation-based correlation reduction mechanism in the sample level, described in Section 3.3 to solve the representation collapse issue in the semi-supervised node classification task.

### 2.3 Interpolation-based Augmentation

Mixup [52, 76] is an effective data augmentation strategy for image classification [69]. It generates synthetic samples by linearly interpolating random image pairs and their labels as follows:

\[
\lambda \sim \text{Beta}(\alpha, \beta), \\
\lambda' = \max(\lambda, 1 - \lambda), \\
x' = \lambda' x_1 + (1 - \lambda') x_2, \\
y' = \lambda' y_1 + (1 - \lambda') y_2, 
\]

(1)

where \(\alpha\) and \(\beta\) are the hyper-parameters of Beta distribution. Besides, \(\lambda \in [0, 1]\) denotes the interpolation rate. Actually, Mixup incorporates the prior knowledge that interpolations of input samples should lead to interpolations of the associated targets [76]. In this manner, it extends the training distribution by constructing virtual training samples across all classes, thus improving the image classification performance [52, 53].

However, it is challenging to extend Mixup methods to the graph data, which contains many irregular connections. To solve this problem, GraphMixup [64] designs feature and edge Mixup mechanisms to improve the performance of class-imbalanced node classification. Besides, Mixup-ForGraph [62] proposed the two-branch graph convolution to mix the receptive field sub-graphs for the paired nodes. Moreover, GraphMix [54] trains a fully connected network (FCN) jointly with the graph neural network (GNN). The interpolated strategy in GraphMix is implemented with the node features \(X\) in FCN. Although the GNN could share parameters with FCN, the topology information still cannot be fully exploited. Different from the previous methods, we propose a simple interpolation fashion. Specially, we interpolate the embeddings and associated labels directly, which can simultaneously learn topology and feature information.

### 3 METHOD

In this section, we proposed a novel graph contrastive learning method, termed Mixed Graph Contrastive Network (MGCN), to improve the latent feature’s discriminative capability and alleviate the collapsed representation. As shown in Figure 2, our proposed method mainly contains two modules, i.e., the graph interpolation module and correlation reduction module. In the
Fig. 2. Illustration of Mixed Graph Contrastive Network (MGCN). In the Graph Interpolation Module, with the generated embedding $H$, we first adopt the interpolation-based strategy to conduct data augmentation in the latent space, and then by guiding $H^{v_2}$ to approximate the prediction $Y^{v_2}$, we force the prediction model to change linearly between samples. Afterward, by guiding the cross-view correlation matrix to approximate the identity matrix, we enable the learned network to tell apart samples across two interpolation-perturbed views. In this manner, our network would be guided to learn the more discriminative embedding, thus alleviating representation collapse. In our model, the interpolation rate $\lambda$ is set as 0.95 to make sure that $H^{v_k}$ is a perturbation of $H$.

following subsections, we first define the main notations and the problem. Then, we detail the two main modules and loss function of MGCN.

3.1 Notations and Problem Definition

To an undirected graph $\{V, E\}$ with $K$ classes of nodes, the node set and the edge set are denoted as $V = \{v_1, v_2, \ldots, v_N\}$ and $E$, respectively. The graph contains an attribute matrix $X \in \mathbb{R}^{N \times D}$ and an adjacency matrix $A = (a_{ij})_{N \times N}$, where $a_{ij} = 1$ if $(v_i, v_j) \in E$, otherwise $a_{ij} = 0$. The degree matrix is denoted as $D = \text{diag}(d_1, \ldots, d_N) \in \mathbb{R}^{N \times N}$ and $d_i = \sum_{(v_i, v_j) \in E} a_{ij}$. The normalized adjacency matrix $\bar{A} \in \mathbb{R}^{N \times N}$ could be calculated through calculating $D^{-\frac{1}{2}}(A + I)D^{-\frac{1}{2}}$, where $I \in \mathbb{R}^{N \times N}$ is an identity matrix. Besides, $\| \cdot \|_2$ denotes the $\ell_2$-norm. In this article, our target is to embed the nodes into the latent space and classify them in a semi-supervised manner. The notations are summarized in Table 1.

3.2 Graph Interpolation Module

Recent works [76] demonstrate that Mixup is an effective data augmentation for images to improve the discriminative capability of samples by achieving larger margin-decision boundaries. Different from images, the nodes in the graph are irregularly connected. Thus, the interpolation for the graph data is still an open question [54, 64].

To overcome this issue, we propose a simple yet effective interpolation method on graph data as shown in the orange box in Figure 2. Specifically, we first encode the nodes into the latent space through Equation (2).

$$H = \mathcal{F}(X, A).$$
Table 1. Notation Summary

| Notations | Meaning |
|-----------|---------|
| $X \in \mathbb{R}^{N \times D}$ | The Attribute Matrix |
| $A \in \mathbb{R}^{N \times N}$ | The Adjacency Matrix |
| $D \in \mathbb{R}^{N \times N}$ | The Degree Matrix |
| $I \in \mathbb{R}^{N \times N}$ | The Identity Matrix |
| $H \in \mathbb{R}^{N \times D}$ | The Node Embeddings |
| $S \in \mathbb{R}^{N \times N}$ | The Cross-view Sample Correlation Matrix |
| $\hat{Y} \in \mathbb{R}^{N \times C}$ | The Prediction Distribution |
| $Y \in \mathbb{R}^{N \times C}$ | The Label Distribution |

Here, $F(\cdot)$ denotes the encoder of our feature extraction framework. In our article, we take the encoder of GPRGNN [8], which learns node embeddings from node features and topological information for sample embedding.

Subsequently, we adopt a simple linear interpolation function $G_k(\cdot)$ to mix the node embeddings as formulated:

$$H^{v_k} = G_k(H) = \lambda H + (1 - \lambda) S_k(H),$$

(3)

where $H^{v_k}$ denotes the $k$th view of the node embedding, and $\lambda = 0.95$ is the interpolation rate. $S_k(\cdot)$ is the shuffle function that randomly permutes the input of the function and output the same samples with a new order. As $\lambda = 0.95$, the interpolation function could be regarded as an operation that introduces perturbation to the principal embedding $H$. Similar to Equation (3), the interpolated labels can be formulated as:

$$Y^{v_k} = G_k(Y) = \lambda Y + (1 - \lambda) S_k(Y).$$

(4)

In this manner, we construct two perturbations $\{H^{v_k}, Y^{v_k}\}$ as two different views of the principal sample batch in the latent space by mixing the node embeddings and the corresponding labels. Subsequently, we enhance the discriminative capability of the network by forcing the prediction model to change linearly between samples through the classification loss:

$$\mathcal{L}_C = CE(\hat{Y}^{v_k}, Y^{v_k}),$$

(5)

where $CE(\cdot)$ denotes the Cross-entropy loss [36] and $\hat{Y}^{v_k}$ is the prediction of training data. According to References [52, 76], in image classification applications, the decision boundaries are pushed far away from the class boundaries by enabling the network to recognize the interpolation operation. Through minimizing $\mathcal{L}_C$ in our article, we can also acquire the larger-margin decision boundaries shown in Figure 3, thus alleviating the representation collapse problem.

3.3 Correlation Reduction Module

To further improve the discriminative capability of samples, we improve the cross-view consistency of the latent representation. Following this idea, as shown in the red box in Figure 2, we propose a correlation reduction module, which pulls together the same samples while pushing away different samples from two interpolation-perturbed views. In this way, our network is encouraged to learn more discriminative embeddings, thus avoiding the representation collapse problem.

Concretely, the process of correlation reduction is divided into three steps. First, we utilize the proposed graph interpolation module to construct two interpolation-perturbed views of node embeddings, i.e., $H^{v_1}$ and $H^{v_2}$ in Figure 2.
ALGORITHM 1: Interpolation-based Correlation Reduction Network

**Input:** An undirected graph $G = (X, A)$; Iteration number $t$; Hyper-parameters $\alpha, \lambda$.

**Output:** Class prediction $\hat{Y}$ and the trained network $F(\cdot)$.

1: \textbf{for} $i = 1$ to $t$ \textbf{do}
2: \hspace{1em} Encode the nodes with the feature extraction network $F(\cdot)$ to obtain the node embeddings $H$;
3: \hspace{1em} Utilize the graph interpolation module to construct two interpolation-perturbed embeddings $H^{v_1}$ and $H^{v_2}$;
4: \hspace{1em} Construct the interpolated labels $Y^{v_2}$ with Equation (4);
5: \hspace{1em} Calculate the classification loss $L_C$ with Equation (5);
6: \hspace{1em} Calculate the correlation matrix $S$ with Equation (6);
7: \hspace{1em} Force $S$ to approximate an identity matrix and calculate information correlation reduction loss $L_R$ with Equation (7);
8: \hspace{1em} Update the whole network by minimizing $L$ in Equation (8);
9: \textbf{end for}
10: Output the predicted classification result $\hat{Y}$.
11: \textbf{return} $\hat{Y}$ and $F(\cdot)$

Second, the correlation matrix $S \in \mathbb{R}^{N \times N}$ across two interpolation-perturbed views is calculated as:

$$S_{ij} = \frac{(H^{v_1}_i)(H^{v_2}_j)^T}{||H^{v_1}_i||_2||H^{v_2}_j||_2},$$  \hspace{1em} (6)

where $S_{ij}$ is the cosine similarity between $i$th node embedding of the first view $H^{v_1}$ and $j$th node embedding of the second view $H^{v_2}$.

Furthermore, we force the correlation matrix $S$ to be equal to an identity matrix $I \in \mathbb{R}^{N \times N}$ by minimizing the information correlation reduction loss, which could be presented as:

$$L_R = \frac{1}{N^2} \sum (S - I)^2 = \frac{1}{N} \sum_{i=1}^{N} (S_{ii} - 1)^2 + \frac{1}{N^2 - N} \sum_{i=1}^{N} \sum_{j \neq i} (S_{ij})^2. \hspace{1em} (7)$$

In detail, the first term in Equation (7) forces the diagonal elements of $S$ to 1, which indicates that the embeddings of each node are forced to agree with each other in two views. Besides, the second term in Equation (7) makes the off-diagonal elements of $S$ to approach 0 to push away different nodes across two views.

By this decorrelation operation, we enlarge the distance between different samples in the latent space while preserving the view-invariance latent feature of each sample, thus keeping cross-view consistent of latent representation. Consequently, our network is guided to learn more discriminative features about input samples and further avoid the collapsed representation.

### 3.4 Loss Function

The proposed method MGCN jointly optimizes two losses: the classification loss $L_C$ and the information correlation reduction loss $L_R$. In summary, the objective of MGCN is formulated as:

$$L = L_C + \alpha L_R, \hspace{1em} (8)$$

where $\alpha$ is a tradeoff hyper-parameter. The detailed learning procedure of MGCN is illustrated in Algorithm 1.
3.5 Theoretical Analysis

In this subsection, we analyze the generalization bound of the proposed MGCN based on Rademacher complexity [1].

For the sake of convenience, we give the following notation: Let $G$ is the fixed graph with $n$ nodes. $S$ is the training set from an unknown distribution $D$ where all the samples are i.i.d according to assumption of the previous researches [2]. $\Gamma$ is the set of hyper-parameters. $H_y$ is the distribution-dependent hypothesis space corresponding to the hyper-parameter $y \in \Gamma$, which can be presented as $H_y = \{h_y : (\exists Z \in \mathbb{Z}) [h_y = A_y(Z)]\}$, where $A_y$ is the algorithm to output the hypothesis $h_y$ for a given training dataset $Z \in \mathbb{Z}$. For each $h_y \in H_y$, $h_y(\cdot, G)$ is the GNN network with $G$. $R_n^f(H_y)$ is the Rademacher complexity of the set $\{n(x) \mapsto \ell(h_y(x, G), y) : h_y \in H_y\}$. $E_{(x, y) \sim \mathcal{D}}[\ell(h_y(x; G), y)] = \frac{1}{n} \sum_{i=1}^{n} L(Z_i, h_y)$. Let $c$ be the upper bound on the loss $L$.

**Theorem 3.1.** For any $\delta > 0$ and $y \in \Gamma$, for all $h_y \in H_y$, with the probability at least $1 - \delta/|\Gamma|$, we have:

\[
E_{(x, y) \sim \mathcal{D}}[\ell(h_y(x; G), y)] - E_{(x, y) \sim \mathcal{D}}[\ell(h_y(x; G), y)] 
\leq 2 \sqrt{\frac{2 \ln \Pi_{H_y}(n)}{n}} + c(2 + \alpha) \sqrt{\frac{\ln(|\Gamma|/\delta)}{2n}},
\]

where $\Pi_{H_y}(n)$ denotes the growth function. For each fixed $H_y$, the generalization bound in Theorem 1 goes to zero, since $\Pi_{H_y}(n)/n \to 0$ and $\ln(|\Gamma|/\delta)/n \to 0$ when $n \to \infty$. In conclusion, the generation gap of our model is approximately $O(1/\sqrt{n})$. Therefore, the generalization bound of the proposed MGCN is promised.

Based on Equations (2), (3), (4), (5), we formalize $L_C$ as $L_C(Z, h_y(\cdot; G), \lambda)$. $L(Z, h_y) = L_C(Z, h_y) + \alpha L_R(Z, h_y)$. Denote $c$ as the upper bound on per-sample loss, i.e., $c \geq \ell(h_y(x_i; G), y_i)$. Let $y \in \Gamma$ be fixed. Let $E_{(x, y) \sim \mathcal{D}}[\ell(h_y(x; G), y)] = \frac{1}{n} \sum_{i=1}^{n} L(Z_i, h_y)$. We define $\psi(Z) = \sup_{h_y \in H_y} E_{(x, y) \sim \mathcal{D}}[\ell(h_y(x; G), y)] - E_{(x, y) \sim \mathcal{D}}[\ell(h_y(x; G), y)]$. To prove Theorem 1, we need the following two lemmas:

**Lemma 3.2.** [1] Let $F$ be a class of real-valued function that map from $X$ to $[0, c]$. Let $D$ be a probability distribution on $X \times [0, c]$, and suppose that sample set $X = \{x_1, x_2, \ldots, x_n\}$ is chosen independently according to the distribution $D$. For all $f \in F$, with probability at least $1 - \delta$, we have:

\[
\Phi(Z) \leq 2 R_n(F) + c \sqrt{\frac{\ln(1/\delta)}{2n}},
\]

where $\Phi(Z) = \sup_{f \in F} E_{(x, y) \sim \mathcal{D}}[f] - E_{(x, y) \sim \mathcal{D}}[f]$, $R_n(\cdot)$ is the correspondent Rademacher complexity.

**Lemma 3.3.** [35] Let $H$ be the hypothesis space. The Rademacher complexity $R_n(H)$ and the growth function $\Pi_{H}(n)$ have:

\[
R_n(H) \leq \sqrt{\frac{2 \ln \Pi_{H}(n)}{n}}.
\]

**Proof.** First, we compute the upper bound on $|\psi(Z) - \psi(Z')|$. Here, $Z$ and $Z'$ denote the two training different datasets in one point of an arbitrary index $i_0$, which is formulated by:

\[
\psi(Z) - \psi(Z') \leq \sup_{h_y \in H_y} L(Z, h_y) - L(Z', h_y)
= \sup_{h_y \in H_y} (L_C(Z, h_y) - L_C(Z', h_y))
+ \alpha (L_R(Z, h_y) - L_R(Z', h_y)),
\]

ACM Trans. Knowl. Discov. Data., Vol. 18, No. 7, Article 162. Publication date: June 2024.
where $Z_i = Z'_i$ for all $i \neq i_0$ and $Z_{i0} \neq Z'_{i0}$. $\mathcal{L}_C$ and $\mathcal{L}_R$ denote the classification loss and information correlation reduction loss, respectively. For the first term:

$$\mathcal{L}_C(Z, h_y(\cdot, G), \lambda) - \mathcal{L}_C(Z', h_y(\cdot, G), \lambda) \leq \frac{c(2n - 1)}{n^2} \leq \frac{2c}{n},$$

where $Z$ and $Z'$ have $n^2$ terms and $2n - 1$ different terms, each of which is bounded by the constant $c$. Therefore, for a fixed $G$ and a fixed $h_y$,

$$\mathcal{L}_C(Z, h_y) - \mathcal{L}_C(Z', h_y) \leq \frac{2c}{n}. \quad (14)$$

For the second term,

$$\mathcal{L}_R(Z, h_y) - \mathcal{L}_R(Z', h_y) \leq \frac{c}{n}. \quad (15)$$

Since $\mathcal{L}_C$ is Cross-entropy loss, the upper bound is 1. Besides, $S$ is calculated by the cosine similarity, and the upper bound of $\mathcal{L}_R$ is 4. Moreover, we adopt the tradeoff hyper-parameter $\alpha$ as 0.5. Therefore, the value of the upper bound $c$ is 3. Based on above proof, $\psi(Z) - \psi(Z') \leq \frac{c(2+\alpha)}{n}$. We could obtain the similar bound $\psi(Z') - \psi(Z) \leq \frac{c(2+\alpha)}{n}$. Therefore, for any $\delta > 0$, with Lemma 2, at least the probability $1 - \delta/|\Gamma|$:

$$\psi(Z) \leq 2\mathcal{R}_n(\mathcal{H}_y) + c(2 + \alpha)\sqrt{\frac{\ln(|\Gamma|/\delta)}{2n}}. \quad (16)$$

Furthermore, with Lemma 3, we have:

$$\psi(Z) \leq 2\sqrt{\frac{2\ln\Pi_H(n)}{n}} + c(2 + \alpha)\sqrt{\frac{\ln(|\Gamma|/\delta)}{2n}}. \quad (17)$$

Therefore, we obtain that for any $\delta > 0$ and all $h_y \in \mathcal{H}_y$, with probability at least $1 - \delta$:

$$\mathbb{E}_{(x,y) \sim \mathcal{D}_n}[\ell(h_y(x, G), y)] - \mathbb{E}_{(x,y) \sim \mathcal{D}_n}[\ell(h_y(x, G), y)] \leq 2\sqrt{\frac{2\ln\Pi_H(n)}{n}} + c(2 + \alpha)\sqrt{\frac{\ln(|\Gamma|/\delta)}{2n}}. \quad (18)$$

\[\square\]

4 EXPERIMENT

4.1 Datasets & Metric

To verify the effectiveness of our proposed method, extensive experiments have been conducted on six benchmark datasets, including DBLP\(^1\) ACM\(^2\) AMAP\(^71\), AMAC\(^71\), CITESEER\(^3\) and CORA\(^27, 41\). Detailed dataset statistics are summarized in Table 2. For fairness, we follow GPRGNN\(^8\) and adopt the sparse splitting (2.5%/2.5%/95% for training/validation/test) in the origin literature for all datasets. The classification performance is evaluated by the wide-used accuracy metric.

\(^1\)https://dblp.uni-trier.de
\(^2\)https://dl.acm.org/
\(^3\)http://citeseerx.ist.psu.edu/index
Table 2. Dataset Summary

| Dataset     | Sample | Dimension | Edges   | Classes |
|-------------|--------|-----------|---------|---------|
| DBLP        | 4,507  | 334       | 7,056   | 4       |
| ACM         | 3,025  | 1,870     | 26,256  | 3       |
| Photo       | 7,650  | 745       | 287,326 | 8       |
| Computers   | 13,752 | 767       | 491,722 | 10      |
| Cora        | 2,708  | 1,433     | 5,429   | 7       |
| Citeseer    | 3,327  | 3,703     | 472     | 6       |

4.2 Experiment Setup

All experiments are implemented with one NVIDIA 1080Ti GPU on PyTorch platform. To alleviate the influence of randomness, we run each method 10 times and report the mean values with standard deviations. Besides, to all methods, we train them for 1,000 epochs until convergence. For ACM and DBLP datasets, we adopt the code of compared methods and reproduce the results. For the performance of baselines on other datasets, we reported the corresponding values from GPRGNN [8] directly. In our proposed method, we adopt GPRGNN as our feature extraction backbone network, and our network is trained with the Adam optimizer [19]. Besides, the learning rate is set to 1e-3 for CITESEER, 5e-2 for DBLP, 2e-2 for CORA, and AMAC, 1e-2 for ACM and AMAP, respectively. The interpolation rate $\lambda$ and the tradeoff hyper-parameter $\alpha$ are set to 0.95 and 0.5, respectively. Moreover, the dimension $D$ is set to 64.

4.3 Performance Comparison

To demonstrate the superiority of our method, we conduct performance comparison experiments for our proposed MGCN and 18 baselines, including classical semi-supervised graph methods, unsupervised contrastive methods, and Mixup-enhanced methods.

- Classical semi-supervised graph methods [8, 9, 11, 20, 21, 38, 50, 63, 68] propagate the supervision information from the labeled data to the unlabeled data according to the linkages within the adjacent matrix as guidance for network training.
- We report the results of the unsupervised contrastive methods [12, 48, 51, 82, 83], which design auxiliary tasks for discriminative information exploitation.
- Mixup-based methods [54, 62, 64] improve the robustness and discriminative capability of the aggregated sample embedding over the labeled samples.

From these results in Table 3, we observe and analyze as follows: (1) It could be observed that the classical GCN-based methods are not comparable with our proposed MGCN. Taking the results on CORA dataset, MGCN exceeds GCN [20] by 5.68%. This is because these methods would suffer from the representation collapse problem caused by the inaccurate connections within data in the adjacency matrix. (2) Our MGCN consistently outperforms other contrastive learning methods. We conjecture that those methods fail to exploit abundant intuitive information within the unlabeled data, thus achieving sub-optimal performance. (3) Compared with the Mixup-enhanced methods, MGCN achieves better classification performance. The reason is that those graph-mix-based methods do not consider the contrastive learning method to improve the discriminative capacity in the semi-supervised node classification task. Overall, MGCN alleviates collapsed representations by improving the discriminative capability of the latent space from two aspects. First, we proposed a graph interpolation to force the prediction model to change linearly between samples. The margin of decision boundaries can be enlarged, thus improving the discriminative of the network. Besides,
Table 3. The Average Semi-supervised Classification Performance with \textit{Mean}±\textit{std} on Six Datasets

| Method       | DBLP   | ACM    | Photo  | Computers | Citeseer | Cora  |
|--------------|--------|--------|--------|-----------|----------|-------|
| MLP          | 63.07 ± 2.33 | 71.02 ± 2.21 | 78.69 ± 0.30 | 70.48 ± 0.28 | 52.88 ± 0.51 | 50.34 ± 0.48 |
| GCN-Cheby    | 60.48 ± 0.00 | 79.98 ± 3.07 | 90.09 ± 0.28 | 82.41 ± 0.28 | 65.67 ± 0.38 | 71.39 ± 0.51 |
| GCN          | 67.64 ± 0.38 | 84.95 ± 0.21 | 90.54 ± 0.21 | 82.52 ± 0.32 | 67.30 ± 0.35 | 75.21 ± 0.38 |
| GraphSage    | 29.49 ± 0.03 | 37.65 ± 0.01 | 90.51 ± 0.25 | 83.11 ± 0.23 | 61.52 ± 0.44 | 70.89 ± 0.54 |
| APPNP        | 67.75 ± 0.44 | 74.61 ± 0.67 | 91.11 ± 0.26 | 81.99 ± 0.26 | 68.59 ± 0.30 | 79.41 ± 0.38 |
| JK-Net       | 64.51 ± 0.53 | 81.20 ± 0.11 | 87.70 ± 0.70 | 77.80 ± 0.97 | 60.85 ± 0.76 | 73.22 ± 0.64 |
| GAT          | 68.58 ± 0.42 | 83.88 ± 0.35 | 90.09 ± 0.27 | 81.95 ± 0.38 | 67.20 ± 0.46 | 76.70 ± 0.42 |
| SGC          | 53.66 ± 2.15 | 72.99 ± 2.96 | 83.80 ± 0.46 | 76.27 ± 0.36 | 58.89 ± 0.47 | 70.81 ± 0.67 |
| GPRGNN       | 67.84 ± 0.30 | 80.93 ± 2.26 | 91.93 ± 0.26 | 82.90 ± 0.37 | 67.63 ± 0.38 | 79.51 ± 0.36 |
| AdaGCN       | 69.70 ± 1.35 | 86.09 ± 1.99 | 46.44 ± 3.25 | 39.71 ± 0.77 | 62.58 ± 1.44 | 62.41 ± 1.84 |
| DGI          | 68.90 ± 1.34 | 81.26 ± 1.48 | 83.10 ± 0.50 | 75.90 ± 0.60 | 63.43 ± 2.94 | 73.74 ± 1.43 |
| GCA          | 60.11 ± 1.94 | 79.23 ± 1.73 | 89.98 ± 1.28 | 81.86 ± 1.80 | 66.25 ± 3.94 | 74.49 ± 3.70 |
| GRACE        | 68.88 ± 0.04 | 85.93 ± 0.56 | 90.60 ± 0.03 | 72.76 ± 0.02 | 66.54 ± 0.01 | 78.62 ± 0.62 |
| MVGRL        | 67.89 ± 0.34 | 83.78 ± 0.27 | 79.37 ± 0.03 | 70.22 ± 0.02 | 67.98 ± 0.05 | 78.06 ± 0.07 |
| BGRL         | 68.48 ± 0.56 | 86.04 ± 0.61 | 90.78 ± 0.89 | 81.40 ± 0.25 | 67.78 ± 0.85 | 78.87 ± 0.41 |
| MixupForGraph| 65.51 ± 0.78 | 86.24 ± 0.62 | 89.87 ± 0.10 | 77.30 ± 2.10 | 57.41 ± 0.33 | 67.11 ± 0.63 |
| GraphMix     | 67.98 ± 0.36 | 85.75 ± 0.55 | 90.14 ± 1.81 | 79.24 ± 0.57 | 66.24 ± 0.74 | 78.54 ± 0.32 |
| GraphMixup   | 68.48 ± 1.52 | 86.44 ± 1.25 | 90.74 ± 0.78 | 81.68 ± 0.42 | 67.53 ± 0.87 | 77.16 ± 1.12 |
| MGCN         | Ours    | 70.60 ± 0.76 | 87.88 ± 0.54 | 92.64 ± 0.24 | 83.99 ± 0.90 | 69.18 ± 0.43 | 80.89 ± 0.95 |

The \textcolor{red}{red} and \textcolor{blue}{blue} values indicate the best and the runner-up results, respectively.

Fig. 3. \textit{t}-SNE visualization of seven methods on two datasets. The first row and second row correspond to ACM and DBLP, respectively.

The proposed correlation reduction mechanism further improves the discriminative capability of the embeddings by keeping the cross-view consistency of the latent representations. Consequently, the proposed MGCN alleviates collapsed representations and achieves the top-level performance on six datasets.

Our proposed MGCN aims to alleviate collapsed representations in graph node classification field. The advantages of MGCN can be summarized as follows: (1) Overcoming irregularity and connectivity challenges: The irregularity and connectivity of graph data make it difficult to directly utilize an interpolation-based strategy. To overcome this challenge, we introduce an interpolation-based framework that enables data augmentation on embeddings and associated labels. (2) Simplified augmentation strategy: Instead of relying on complex graph augmentation techniques such as graph diffusion or attribute shuffle, we employ an easily implemented interpolation-based strategy to manipulate the embeddings. This approach proves effective in enlarging the decision boundaries and improving the model’s discriminative capacity.
Fig. 4. Ablation comparisons of the proposed modules on six datasets. “B,” “B+I,” “B+C,” and “Ours” denote the baseline, the baseline with graph interpolation module, correlation reduction module, and both, respectively.

Fig. 5. Testing of the effectiveness and sensitivity of hyper-parameters $\alpha$ and $\lambda$. The result perturbation with the variation of the two parameters on all six datasets are illustrated in the figures.

4.4 Transferring Modules to Other Methods

To further investigate the effectiveness and the generality of our proposed modules, we transfer the graph interpolation module and correlation reduction module to five baselines, including GCN-Cheby [9], GCN [20], APPNP[21], JK-Net [68], GAT [50]. Table 5 reports the performance of the five methods with their variants on DBLP, ACM, CITESEER, and CORA datasets. Here, we denote the baseline and the baseline with the two proposed modules as B and B-O, respectively.

From these results, we observed that, enhanced by our proposed modules, the baselines significantly achieve better performance. Specifically, our modules improve the classification accuracy of GCN by 4.79% on DBLP, 0.82% on ACM, 1.23% on CITESEER, and 2.49% on CORA, respectively. The reason is that the two proposed modules enhance the discriminative capability of samples by conducting the interpolation-based augmentations in the latent space and improving the cross-view consistency of the node representations. In this manner, the baselines alleviate the collapsed representation, thus achieving better classification performance.
Table 4. Time Cost Comparisons of the Training Process

| Method     | Cora | Citeseer | AMAP | AMAC | ACM | DBLP |
|------------|------|----------|------|------|-----|------|
| GCN        | 8.56s| 9.33s    | 21.26s| 37.77s| 8.45s| 8.2s |
| APPNP      | 11.45s| 14.25s  | 12.75s| 38.75s| 11.39s| 11.34s|
| JKNet      | 14.9s | 14.71s   | 15.01s| 57.56s| 14.79s| 14.71s|
| ChebyNet   | 13.9s | 22.49s   | 46.37s| 90.65s| 20.89s| 10.42s|
| MixupForGraph | 13.35s| 20.66s  | 120.1s| 59.68s| 17.23s| 11.47s|
| GPRGNN     | 3.06s | 3.06s    | 3.5s  | 6.05s | 10s  | 10.37s|
| MGCN       | 7.18s | 9.55s    | 15.5s | 49.17s| 7.06s | 6.6s |

All results are measured in seconds with 10 runs.

Table 5. Transferring Our Proposed Modules to Other Models on Four Datasets

| Dataset | GCN-Cheby | GCN | APPNP | JKNet | GAT |
|---------|-----------|-----|-------|-------|-----|
| DBLP    | 60.48 ± 0 | 63.52 ± 1.46 | 65.64 ± 0.38 | 72.43 ± 0.62 | 68.59 ± 0.78 | 65.51 ± 0.53 | 66.97 ± 0.19 | 68.58 ± 0.42 | 69.00 ± 1.84 |
| ACM     | 79.98 ± 3.07 | 83.62 ± 1.03 | 85.95 ± 0.21 | 85.77 ± 1.33 | 74.61 ± 0.67 | 83.71 ± 1.78 | 81.20 ± 0.11 | 83.53 ± 1.22 | 83.88 ± 0.35 | 83.18 ± 2.95 |
| CITESEER| 65.67 ± 0.38 | 66.62 ± 0.65 | 67.30 ± 0.35 | 68.53 ± 0.59 | 68.59 ± 0.30 | 70.12 ± 0.97 | 64.51 ± 0.56 | 64.88 ± 1.00 | 67.20 ± 0.46 | 68.54 ± 0.38 |
| CORA    | 71.39 ± 0.51 | 72.93 ± 1.06 | 73.21 ± 0.35 | 77.30 ± 0.44 | 79.41 ± 0.35 | 79.53 ± 0.37 | 73.22 ± 0.84 | 75.45 ± 1.64 | 67.70 ± 0.42 | 77.28 ± 3.25 |

“B” and “B-O” represent the baseline and the baseline with our method, respectively. Boldface letters are used to mark the best results.

4.5 Time Cost and Memory Cost

In this subsection, we implement time and memory cost experiments to demonstrate the effectiveness of the proposed MGCN. Specifically, we test the training time of MGCN with five baselines on six datasets. For fairness, we train all algorithms with 1,000 epochs. The results are shown in Table 4. From the results, we observe that the training time of MGCN is comparable with other state-of-the-art methods. The reasons we analyze are as follows: (1) Following GPRGNN [8], instead of using GCN, we adopt generalized pagerank (GPR) to jointly extract node feature and topological information. This operation effectively reduces time consumption. (2) MGCN utilizes Equation (6) as loss function. Compared with commonly contrastive loss function, e.g., temperature-scaled cross-entropy loss (NT-Xent) [7, 65, 69], the loss function in MGCN is more efficient. The detailed description is shown in Section 4.6.3.

Moreover, we conduct experiments to test GPU memory costs of our proposed MGCN with six methods (i.e., MixupForGraph [62], GCN [20], GPRGNN [8], APPNP [21], JKNet [68], ChebNet [9]) on six datasets. From the results in Figure 7, we observe that the memory costs of our MGCN are comparable with other algorithms. We summarize the reasons as follows: (1) Following GPRGNN [8], MGCN adopts the simple network architecture. (2) Compared with temperature-scaled cross-entropy loss (NT-Xent) [7, 69] in other contrastive learning algorithms, MGCN utilizes MSE loss to reduce memory costs.

4.6 Ablation Studies

In this section, we first conduct ablation studies to verify the effectiveness of the proposed modules, and then we analyze the robustness of MGCN to the hyper-parameters. Last, we conduct experiments to verify the effectiveness of our proposed loss function.

4.6.1 Effectiveness of the Proposed Modules. To investigate the effectiveness of the proposed graph interpolation module and correlation reduction module, extensive ablation studies are conducted in Figure 4. Here, we adopt GPNGNN [8] as “Baseline.” Besides, “B,” “B+I,” “B+C,” and “Ours” denote the baseline, the baseline with graph interpolation module, correlation reduction module, and both, respectively. From these results, we have observed as follows: (1) Compared with “Baseline,” “B+I” has about 1.81% performance improvement on average of six datasets, since the
proposed graph interpolation module enlarges the margin of decision boundaries by forcing the prediction model to change linearly between samples. (2) Benefitting from the correlation reduction module, the classification performance is improved. Taking the result on DBLP dataset for example, “B+C” exceeds “Baseline” by 2.05%. This demonstrates that the correlation reduction module improves the discriminative capability of samples by keeping the cross-view consistency of the latent representations. (3) Moreover, better performance of “Ours” indicates that both proposed modules are effective to guide the network to learn more discriminative latent features.

4.6.2 Hyper-parameter Analysis. Furthermore, we investigate the robustness of our proposed method to the hyper-parameters on six datasets. Specifically, to the tradeoff hyper-parameter $\alpha$, we conduct ablation studies as shown in Figure 5(a). From these results, we observe that the classification accuracy will not fluctuate greatly when $\alpha \in [0.3, 0.7]$. This demonstrates that our model MGCN is insensitive when $\alpha$ varying in a tiny range. Besides, the accuracy of semi-supervised node classification with different values of the interpolation rate $\lambda$ are illustrated in Figure 5(b). It
Mixed Graph Contrastive Network for Semi-Supervised Node Classification

Fig. 8. Time costs and GPU memory costs of NT-Xent and our loss functions on five datasets. NT-Xent denotes temperature-scaled cross-entropy loss.

is observed that the performance of MGCN is decreased when $\lambda$ is about less than 0.95, since $\lambda$ controls the perturbation to the principal embedding $H$. It is worth mentioning that $\lambda$ is set as 0.95 in all experiments.

4.6.3 Effectiveness of the Loss Function. To demonstrate the effectiveness of our proposed loss function, we have conducted experiments to test the GPU memory costs and time costs. Specifically, we compare with NT-Xent loss [7] on five datasets, including CORA, CITESEER, AMAP, ACM, and DBLP. From Figure 8, we could observe that the time cost of our loss function is less than NT-Xent. Moreover, the GPU memory costs are also comparable.

4.7 Visualization Experiment

4.7.1 t-SNE Visualization of Classification Results. To intuitively show the superiority of MGCN, we visualize the distribution of the node embeddings $H$ learned by ChebNet, GCN, MV-GRL, GPRGNN, and our MGCN on ACM and DBLP datasets via t-SNE algorithm [49]. Here, we randomly select two categories of all samples to illustrate the margin of the corresponding decision boundaries clearly in Figure 3. From these results, we conclude that our proposed method has a larger margin of the decision boundaries compared with others.

4.7.2 Visualization of Node Similarity Matrices. We plot the heat maps of sample similarity matrices in the latent space to intuitively show the representation collapse problem in graph node classification methods and the effectiveness of our solution to this issue on DBLP and AMAP datasets. Here, we sort all samples by categories to make those from the same cluster beside each other. As illustrated in Figure 6, we observe that GCN [20] and GPRGNN [8] would suffer from representation collapse during the process of node encoding. Unlike them, our proposed method learns the more discriminative latent features, thus avoiding the representation collapse.

5 CONCLUSION

In this work, we propose a novel graph contrastive learning method termed Interpolation-based Correlation Reduction Network (MGCN) to alleviate the representation issue in semi-supervised node classification task. Specifically, we propose a graph interpolation module to force the prediction model to change linearly between samples, thus enlarging the margin of decision boundaries. Besides, the proposed correlation reduction module aims to keep the cross-view consistency of the embeddings. Benefitting from these two modules, our network is guided to learn more discriminative representations, thus alleviating the representation collapse problem. Extensive experiments
on six datasets demonstrate the superiority of our proposed methods. In MGCN, the designed augmentation strategy interpolates the embeddings and associated labels. As a future work direction, it would be meaningful to explore how to adapt the interpolation-based augmentation strategy to unsupervised tasks.

REFERENCES

[1] Peter L. Bartlett and Shahar Mendelson. 2002. Rademacher and Gaussian complexities: Risk bounds and structural results. *J. Mach. Learn. Res.* 3, Nov. (2002), 463–482.

[2] Christopher Beckham, Sina Honari, Vikas Verma, Alex M. Lamb, Farnoosh Ghadiri, R. Devon Hjelm, Yoshua Bengio, and Chris Pal. 2019. On adversarial mixup resynthesis. *Adv. Neural Inf. Process. Syst.* 32 (2019).

[3] Piotr Bielak, Tomasz Kajdanowicz, and Nitesh V. Chawla. 2021. Graph Barlow twins: A self-supervised representation learning framework for graphs. *arXiv preprint arXiv:2106.02466* (2021).

[4] Jin Chen, Defu Lian, Binbin Jin, Xu Huang, Kai Zheng, and Enhong Chen. 2022. Fast variational autoencoder with inverted multi-index for collaborative filtering. In *Proceedings of the ACM Web Conference*. 1944–1954.

[5] Jin Chen, Defu Lian, Yucheng Li, Baoyun Wang, Kai Zheng, and Enhong Chen. 2022. Cache-augmented inbatch importance resampling for training recommender retriever. *Adv. Neural Inf. Process. Syst.* 35 (2022), 34817–34830.

[6] Jin Chen, Guanyu Ye, Yan Zhao, Shuncheng Liu, Liwei Deng, Xu Chen, Rui Zhou, and Kai Zheng. 2022. Efficient join order selection learning with graph-based representation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 97–107.

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the International Conference on Machine Learning*.

[8] Eli Chien, Jianhao Peng, Pan Li, and Olga Milenkovic. 2020. Adaptive universal generalized pagerank graph neural network. *arXiv preprint arXiv:2006.07988* (2020).

[9] M. Defferrard, X. Bresson, and P. Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in Neural Information Processing Systems* 29 (2016).

[10] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Irena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Ic Hal Valko. 2020. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733* (2020).

[11] William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Proceedings of the International Conference on Neural Information Processing (ICONIP’17)*.

[12] Kaveh Hassani and Amir Hosein Khasahmadi. 2020. Contrastive multi-view representation learning on graphs. In *Proceedings of the International Conference on Machine Learning (ICML’20)*.

[13] Rundong He, Zhongyi Han, Xiankai Lu, and Yilong Yin. 2022. Safe-student for safe deep semi-supervised learning with unseen-class unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 14585–14594.

[14] Rundong He, Zhongyi Han, Yang Yang, and Yilong Yin. 2022. Not all parameters should be treated equally: Deep safe semi-supervised learning under class distribution mismatch. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 6874–6883.

[15] Jincheng Huang, Lun Du, Xu Chen, Qiang Fu, Shi Han, and Dongmei Zhang. 2023. Robust mid-pass filtering graph convolutional networks. In *Proceedings of the ACM Web Conference*. 328–338.

[16] Jincheng Huang, Ping Li, Rui Huang, Na Chen, and Acong Zhang. 2023. Revisiting the role of heterophily in graph representation learning: An edge classification perspective. *ACM Trans. Knowl. Discov. Data* 18, 106 (2023), 1–17. https://doi.org/10.1145/3603378

[17] Jincheng Huang, Pin Li, and Kai Zhang. 2022. Semantic consistency for graph representation learning. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN’22)*. IEEE, 1–8.

[18] Baoyu Jing, Chanyoung Park, and Hanghang Tong. 2021. HDMI: High-order Deep Multiplex Infomax. In *Proceedings of the Web Conference*. 2414–2424.

[19] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[20] Thomas N. Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[21] Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. 2018. Predict then propagate: Graph neural networks meet personalized pagerank. *arXiv preprint arXiv:1810.05997* (2018).

[22] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE Conference*.
[23] Meng Liu, Ke Liang, Yawei Zhao, Wenxuan Tu, Sihang Zhou, Xinwang Liu, and Kunlun He. 2023. Self-supervised temporal graph learning with neural and structural intensity alignment. *arXiv preprint arXiv:2302.07491* (2023).

[24] Suyuan Liu, Xinwang Liu, Siwei Wang, Xin Niu, and En Zhu. 2022. Fast incomplete multi-view clustering with view-independent anchors. *IEEE Trans. Neural Netw. Learn. Syst.* (2022). DOI: [10.1109/TNNLS.2022.3220486](https://doi.org/10.1109/TNNLS.2022.3220486).

[25] Suyuan Liu, Siwei Wang, Pei Zhang, Kai Xu, Xinwang Liu, Changwang Zhang, and Feng Gao. 2022. Efficient one-pass multi-view subspace clustering with consensus anchors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 7576–7584.

[26] Yue Liu, Ke Liang, Jun Xia, Sihang Zhou, Xihong Yang, Xinwang Liu, and Z. Stan Li. 2023. Dink-Net: Neural clustering on large graphs. In *Proceedings of the International Conference on Machine Learning (ICML’23)*.

[27] Yue Liu, Wenxuan Tu, Sihang Zhou, Xinwang Liu, Limxuan Song, Xihong Yang, and En Zhu. 2022. Deep graph clustering via dual correlation reduction. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI’22)*.

[28] Yue Liu, Xihong Yang, Sihang Zhou, Xinwang Liu, Siwei Wang, Ke Liang, Wenxuan Tu, and Liang Li. 2023. Simple contrastive graph clustering. *IEEE Trans. Neural Netw. Learn. Syst.* (2023). DOI: [10.1109/TNNLS.2023.3271871](https://doi.org/10.1109/TNNLS.2023.3271871).

[29] Yue Liu, Xihong Yang, Sihang Zhou, Xinwang Liu, Zhen Wang, Ke Liang, Wenxuan Tu, Liang Li, Jingcan Duan, and Cancan Chen. 2023. Hard sample aware network for contrastive deep graph clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI’23)*, Vol. 37. 8914–8922.

[30] Yue Liu, Sihang Zhou, Xinwang Liu, Wenxuan Tu, and Xihong Yang. 2022. Improved dual correlation reduction network. *arXiv preprint arXiv:2202.12533* (2022).

[31] Zepu Lu, Defu Lian, Jin Zhang, Zaxi Zhang, Chao Feng, Hao Wang, and Enhong Chen. 2023. Differentiable optimized product quantization and beyond. In *Proceedings of the ACM Web Conference*. 3353–3363.

[32] Yujie Mo, Yuhuan Chen, Yajie Lei, Liang Peng, Xiaoshuang Shi, Changan Yuan, and Xiaofeng Zhu. 2023. Multiplex graph representation learning via dual correlation reduction. *IEEE Trans. Knowl. Data Eng.* 35, 12 (2023), 1–14. DOI: [https://doi.org/10.1109/TKDE.2023.3268069](https://doi.org/10.1109/TKDE.2023.3268069).

[33] Yujie Mo, Yajie Lei, Jialie Shen, Xiaoshuang Shi, Heng Tao Shen, and Xiaofeng Zhu. 2023. Disentangled multiplex graph representation learning. In *Proceedings of the 40th International Conference on Machine Learning*.

[34] Yujie Mo, Liang Peng, Jie Xu, Xiaoshuang Shi, and Xiaofeng Zhu. 2022. Simple unsupervised graph representation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI’22)*. 7797–7805.

[35] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. 2018. *Foundations of Machine Learning*. MIT Press.

[36] Kevin P. Murphy. 2012. *Machine Learning: A Probabilistic Perspective*. MIT Press.

[37] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1998. The pagerank citation ranking: Bringing order to the web. Technical report[J]. *Stanford Digital Library Technologies Project*.

[38] S. K. Pal and S. Mitra. 1992. Multilayer perceptron, fuzzy sets, and classification. *IEEE Transactions on Neural Networks* 3, 5 (1992), 683–697.

[39] Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. 2020. Geom-GCN: Geometric graph convolutional networks. *arXiv preprint arXiv:2002.05287* (2020).

[40] Zhen Peng, Wenbing Huang, Minnan Luo, Qinghua Zheng, Yu Rong, Tingyang Xu, and Junzhou Huang. 2020. Graph representation learning via graphical mutual information maximization. In *Proceedings of the Web Conference*. 259–270.

[41] Oleksandr Shchur, Maximilian Mumme, Aleksandr Bojchevski, and Stephan Günnemann. 2018. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868* (2018).

[42] Fan-Yun Sun, Jordan Hoffmann, Vikas Verma, and Jian Tang. 2019. InfoGraph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. *arXiv preprint arXiv:1908.01000* (2019).

[43] Ke Sun, Zhanxing Zhu, and Zhouchen Lin. 2019. AdaGCN: Adaboosting graph convolutional networks into deep models. *arXiv preprint arXiv:1908.05081* (2019).

[44] Cheng Tan, Zhangyang Gao, Siyuan Li, and Stan Z. Li. 2022. SimVP: Towards simple yet powerful spatiotemporal predictive learning. *arXiv preprint arXiv:2211.12509* (2022).

[45] Cheng Tan, Zhangyang Gao, Lirong Wu, Siyuan Li, and Stan Z. Li. 2022. Hyperspherical consistency regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7244–7255.

[46] Cheng Tan, Siyuan Li, Zhangyang Gao, Wenfei Guan, Zedong Wang, Zicheng Liu, Lirong Wu, and Stan Z. Li. 2023. OpenSTL: A comprehensive benchmark of spatio-temporal predictive learning. In *Proceedings of the Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

[47] Shantanu Thakoor, Corentin Tallec, Mohammad Gheshtlaghi Azar, Mehdi Azabou, Eva L. Dyer, Remi Munos, Petar Velicković, and Michal Valko. 2021. Large-scale representation learning on graphs via bootstrapping. In *Proceedings of the International Conference on Learning Representations*.

[48] Shantanu Thakoor, Corentin Tallec, Mohammad Gheshtlaghi Azar, Mehdi Azabou, Eva L. Dyer, Remi Munos, Petar Velicković, and Michal Valko. 2021. Large-scale representation learning on graphs via bootstrapping. *arXiv preprint arXiv:2102.06514* (2021).
[73] Xihong Yang, Cheng Tan, Yue Liu, Ke Liang, Siwei Wang, Sihang Zhou, Jun Xia, Stan Z. Li, Xinwang Liu, and En Zhu. 2023. Convert: Contrastive graph clustering with reliable augmentation. In Proceedings of the 31st ACM International Conference on Multimedia. 319–327.

[74] Mingjia Yin, Hao Wang, Xiang Xu, Likang Wu, Sirui Zhao, Wei Guo, Yong Liu, Ruiming Tang, Defu Lian, and Enhong Chen. 2023. APGL4SR: A generic framework with adaptive and personalized global collaborative information in sequential recommendation. In Proceedings of the 32nd ACM International Conference on Information and Knowledge Management. 3009–3019.

[75] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph contrastive learning with augmentations. Adv. Neural Inf. Process. Syst. 33 (2020), 5812–5823.

[76] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. 2017. Mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412 (2017).

[77] Zaixi Zhang and Qi Liu. 2023. Learning subpocket prototypes for generalizable structure-based drug design. In Proceedings of the International Conference on Machine Learning (ICML ’23).

[78] Z. Zhang, Z. Lu, H. Zhongkai, et al. 2024. Full-atom protein pocket design via iterative refinement. Advances in Neural Information Processing Systems 36 (2024).

[79] Chuang Zhao, Hongke Zhao, Ming He, Jian Zhang, and Jianping Fan. 2023. Cross-domain recommendation via user interest alignment. In Proceedings of the ACM Web Conference. 887–896.

[80] Chuang Zhao, Hongke Zhao, Xiaomeng Li, Ming He, Jiahui Wang, and Jianping Fan. 2023. Cross-domain recommendation via progressive structural alignment. IEEE Trans. Knowl. Data Eng. (2023). DOI: 10.1109/TKDE.2023.3324912

[81] Qun Zheng, Xihong Yang, Siwei Wang, Xinru An, and Qi Liu. 2023. Asymmetric double-winged multi-view clustering network for exploring diverse and consistent information. arXiv preprint arXiv:2309.00474 (2023).

[82] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2020. Deep graph contrastive representation learning. In Proceedings of the ICML Workshop on Graph Representation Learning and Beyond. Retrieved from http://arxiv.org/abs/2006.04131

[83] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2021. Graph contrastive learning with adaptive augmentation. In Proceedings of the Web Conference. 2069–2080.

[84] Zhihao Zhu, Chenwang Wu, Rui Fan, Defu Lian, and Enhong Chen. 2023. Membership inference attacks against sequential recommender systems. In Proceedings of the ACM Web Conference 2023. 1208–1219.

[85] Zhihao Zhu, Chenwang Wu, Min Zhou, Hao Liao, Defu Lian, and Enhong Chen. 2022. Resisting graph adversarial attack via cooperative homophilous augmentation. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, 251–268.

Received 18 August 2023; revised 12 December 2023; accepted 8 January 2024