Improved Dual Correlation Reduction Network

Journal:	*Transactions on Pattern Analysis and Machine Intelligence*
Manuscript ID	TPAMI-2022-02-0371
Manuscript Type:	Regular
Keywords:	Deep Graph Clustering, Graph Neural Network, Self-Supervised Learning, Representation Collapse

Improved Dual Correlation Reduction Network

Yue Liu*, Sihang Zhou*, Xinwang Liu†, *Senior Member, IEEE*, Wenxuan Tu, Xihong Yang,
Xin Xu, *Senior Member, IEEE*, Fuchun Sun, *Fellow, IEEE*

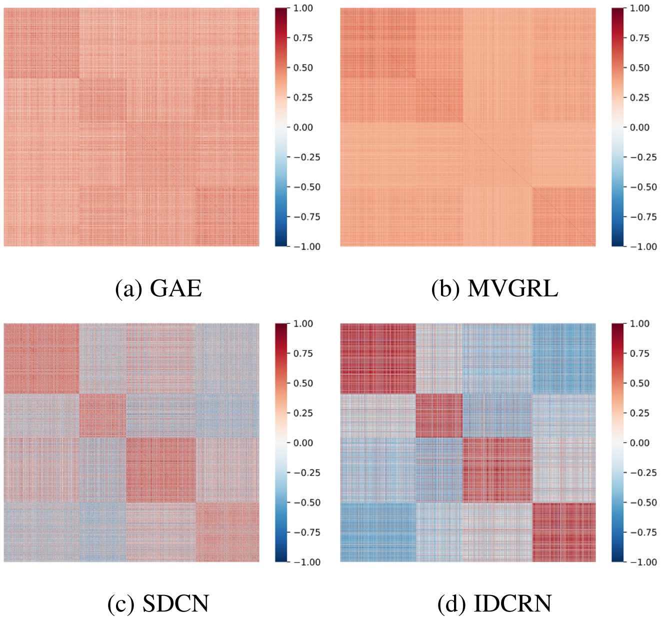
Abstract—Deep graph clustering, which aims to reveal the underlying graph structure and divide the nodes into different clusters without human annotations, is a fundamental yet challenging task. However, we observed that the existing methods suffer from the representation collapse problem and easily tend to encode samples with different classes into the same latent embedding. Consequently, the discriminative capability of nodes is limited, resulting in sub-optimal clustering performance. To address this problem, we propose a novel deep graph clustering algorithm termed Improved Dual Correlation Reduction Network (IDCRN) through improving the discriminative capability of samples. Specifically, by approximating the cross-view feature correlation matrix to an identity matrix, we reduce the redundancy between different dimensions of features, thus improving the discriminative capability of the latent space explicitly. Meanwhile, the cross-view sample correlation matrix is forced to approximate the designed clustering-refined adjacency matrix to guide the learned latent representation to recover the affinity matrix even across views, thus enhancing the discriminative capability of features implicitly. Moreover, we avoid the collapsed representation caused by the over-smoothing issue in Graph Convolutional Networks (GCNs) through an introduced propagation regularization term, enabling IDCRN to capture the long-range information with the shallow network structure. Extensive experimental results on six benchmarks have demonstrated the effectiveness and the efficiency of IDCRN compared to the existing state-of-the-art deep graph clustering algorithms.

Index Terms—Deep Graph Clustering, Graph Neural Network, Self-Supervised Learning, Representation Collapse.

1 Introduction

Deep graph clustering is a fundamental yet challenging task that aims to reveal the underlying graph structure and divide the nodes into different clusters in a self-supervised manner. GCN [1], which possesses the powerful graph representation learning capability, has achieved promising performance in the attribute graph clustering task. Consequently, based on GCN, many deep graph clustering methods are proposed [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13].

Although promising performance has been achieved, we observed that the existing deep graph clustering algorithms [9], [14], [15] suffer from the representation collapse issue [16] and easily embed the nodes from different classes into the same embedding. To solve this problem, many attempts have been made. The contrastive-strategy-enhanced method [15] pushes negative sample pairs away while pulling positive sample pairs together, which to some extent alleviating the collapsed representations. However, since the definition of positive and negative sample pairs is not accurate enough, the learned latent features are still indiscriminative. SDCN [9] exploits both the structural information and attribute information with an information transport operation, which alleviates the over-smoothing problem. However, SDCN [9] neglects the correlation of the latent embeddings, thus leading to sub-optimal clustering performance. We conduct a simple experiment on the DBLP dataset to illustrate this issue. Specifically, in the experiment, we first extract the node embeddings of four well-trained deep graph clustering algorithms, i.e., GAE [14], MVGRL [15], SDCN [9], and our proposed algorithm (IDCRN). Afterward, we calculate the cosine similarity between the embedded samples in the latent space and visualize the resultant similarity matrix for each algorithm. From Fig. 1 (a-c), we observe that although the



(a) GAE (b) MVGRL

(c) SDCN (d) IDCRN

Figure 1: Visualization of the sample similarity matrices in the latent space learned by the four compared algorithms, i.e., GAE [14], MVGRL [15], SDCN [9], and our proposed method (IDCRN), on the DBLP dataset. The sample order is rearranged to make those from the same cluster beside each other.

- *Y. Liu, X. Liu, W. Tu, X. Yang are with School of Computer, National University of Defense Technology, Changsha, 410073, China. (E-mail: {yueliu, xinwangliu, twx, yangxihong } @nudt.edu.cn)*
- *S. Zhou and X. Xin are with College of Intelligence Science and Technology, National University of Defense Technology, Changsha 410073, China.*
- *F. Sun is with Department of Computer Science and Technology, Tsinghua University, Beijing, China.*
- **: Equal contribution. †: Corresponding author.*
- *Manuscript received Feb. 23, 2022.*

problem of over-smoothing is alleviated to different extents, the intrinsic four-dimensional cluster structure is not well revealed by the compared three algorithms [9], [14], [15]. This phenomenon illustrates that the representation collapse problem is still an open problem that limits the clustering performance of the existing deep graph clustering algorithms.

To address the representation collapse problem, we propose a novel contrastive-learning-based deep graph clustering network termed Improved Dual Correlation Reduction Network (IDCRN) by improving the discriminative capability of nodes. In our algorithm, the discriminative capability is improved in two aspects, i.e., the feature aspect and the sample aspect. Specifically, to the feature aspect, we first construct two augmented graph views and encode the nodes with a siamese network. Subsequently, we reduce the redundancy between different dimensions of features via approximating the cross-view feature correlation matrix to the identity matrix. With this setting, the discriminative capability of the latent space is enhanced explicitly, thus alleviating the collapsed representation. Moreover, in the sample aspect, we force the cross-view sample correlation matrix to approximate the high confident clustering results refined affinity matrix. In this manner, we guide the learned latent representation to recover the affinity matrix even across views, thus improving the feature discriminative capability implicitly. As shown in Fig. 1 (d), we found that, with our improved dual correlation reduction module, IDCRN could better reveal the latent cluster structure among data than the other compared algorithms [9], [14], [15]. Besides, IDCRN saves more GPU space than the contrastive-learning-based algorithms since it eliminates the space-consuming negative sample generation operation. For instance, our proposed method saves about 54% GPU memory on average compared to MVGRL [15] during training on DBLP, CITE, and ACM datasets. Moreover, motivated by the Propagation-regularization (P-reg) [17], we avoid the collapsed representations caused by over-smoothing in GCN by a propagation-regularization term, thus further improving the clustering performance of the proposed method.

This work is an extended version of our AAAI 2022 conference paper, i.e., dual correlation reduction network (DCRN) [18]. In our conference paper, both the sample-level and feature-level cross-view correlation matrices are forced to approximate the identity matrices. However, through our observation and experimental validation, we found that, to improve the discriminative capability, instead of just guiding the network to tell apart samples across views, guiding the network to reveal the underlying sample distribution would endow it with better discriminative capability. To this end, we approximate the cross-view sample correlation matrix to a designed clustering-refined affinity matrix instead of the identity matrix. In this manner, the discriminative capability of latent features wound be enhanced, thus further improving the clustering performance. Take the clustering results on DBLP dataset as an example, our proposed algorithm improves 5.21% on the metric of ARI compared to DCRN [18]. Moreover, more sufficient experimental studies are conducted to demonstrate the effectiveness and efficiency of the proposed algorithm. Three contributions of this work are listed as follows.

- A novel contrastive-learning-based method termed IDCRN is proposed to solve the representation collapse problem in the existing deep graph clustering methods.
- We propose two strategies to enhance the discriminative capability of samples implicitly and explicitly. Besides, compared to other contrastive-learning-based methods, IDCRN saves more GPU memory since it gets rid of the complicated negative sample generation operation.
- More sufficient experimental results on six benchmarks have verified the superiority of IDCRN compared to the existing state-of-the-art deep graph clustering methods.

2 Related Works

2.1 Deep Graph Clustering

Graph Convolutional Networks (GCNs), which possess the powerful graph representation learning capability, have achieved impressive performance in the field of deep graph clustering. Specifically, the authors of GAE / VGAE [14] firstly design a graph encoder to learn the node embeddings from both the attributes and the structure information and then reconstruct the adjacency matrix by an inner product decoder. Based on GAE / VGAE, recent researches, including DAEGC [2] and GALA [6] improve the clustering performance with the attention mechanism and the Laplacian sharpening, respectively. Moreover, other two methods termed ARGA / ARVGA [3] and AGAE [4] enhance the discriminative capability of samples by generative adversarial learning, thus achieving the promising clustering performance. Though ahead mentioned works have improved the clustering performance of early works, the over-smoothing problem is still not solved. SDCN / SDCN$_Q$ [9] and DFCN [11] are proposed to jointly train an AE [19] and a GAE [14] to avoid the over-smoothing issue through the designed information transport operation and the attribute-structure fusion strategy, respectively. Similar to DFCN [11], AGCN [10] also demonstrates the effectiveness of the attribute-structure fusion module. More recently, the contrastive-learning-based methods including MCGC [20] and MVGRL [15] aim to learn consensus node embeddings from different views of the graph by introducing the contrastive loss, thus further improving the clustering performance. Although recent works have proved the effectiveness of learning consensus information from different views of the graph data, we found that they suffer from the representation collapse problem and easily map samples from different classes into the same embedding, leading to the unsatisfied clustering performance. To avoid the collapsed representation, we propose a novel contrastive-learning-based deep graph clustering method by improving the discriminative capability of the learned embeddings in the sample and feature aspects.

2.2 Representation Collapse

Representation collapse is a common problem that the network tends to encode samples from different classes into the same representation in the field of the self-supervised learning [16]. Contrastive learning [21] is one possible way to solve this problem. Specifically, a pioneer termed MoCo [22] adopts the momentum encoder to keep the consistency of the negative pair embeddings from the designed memory bank. After that, another effective contrastive learning method termed SimCLR [23], defines the “negative” and “positive” sample pairs and then pushes the “negative” samples away while pulling closer the “positive” samples. Subsequently, BYOL [24] introduces three strategies, including momentum encoder, predictor, and gradient stopping to address this problem. Different from them, a scalable online clustering method named SwAV [25] alleviates the collapsed representations by mapping the representations into different clusters. Moreover, SimSiam [26] has demonstrated that the stop-gradient mechanism is crucial to alleviate the collapsed representations without negative samples. More recently, Barlow Twins [27] and VICREG [28] propose the effective yet simple redundancy reduction mechanisms to alleviate the representation collapse issue even without the momentum encoder, the negative sample pairs, or the gradient stopping mechanism. Motivated by their success [27], [28], DCRN [18] is designed to solve the representation collapse problem through reducing the correlation in the sample and feature aspects. Following DCRN [18], we further enhance the discriminative capability of learned node embeddings by guiding the network to reveal the underlying sample distribution, thus further improving the clustering performance.

3 Proposed Method

In this section, we propose a novel contrastive-learning-based graph clustering method termed Improved Dual Correlation Reduction Network (IDCRN) to solve the representation collapse issue by improving the discriminative capability of the node embeddings. Fig. 2 illustrates that IDCRN mainly contains two components including graph augmentation module, and Improved Dual Correlation Reduction Module (IDCRM). In what follows, we first define the notations and formulate the problem. Subsequently, we will detail graph augmentation module, IDCRM, and the overall objective function.

3.1 Notations Definition and Problem Formulation

Let $\mathcal{V} = \{v_1, v_2, \dots, v_N\}$ be a set of N nodes with C classes and $\mathcal{E}$ be a set of edges. In the matrix form, $\mathbf{X} \in \mathbb{R}^{N\times D}$ and $\mathbf{A} \in \mathbb{R}^{N\times N}$ denote the node attribute matrix and the original adjacency matrix, respectively. Then $\mathcal{G} = \{\mathbf{X}, \mathbf{A}\}$ denotes an undirected graph. The degree matrix is formulated as $\mathbf{D} = diag(d_1, d_2, \dots, d_N) \in \mathbb{R}^{N\times N}$ and $d_i = \sum_{(v_i,v_j)\in\mathcal{E}} a_{ij}$. The normalized adjacency matrix $\widetilde{\mathbf{A}} \in \mathbb{R}^{N\times N}$ is calculated by $\hat{\mathbf{D}}^{-\frac{1}{2}}\hat{\mathbf{A}}\hat{\mathbf{D}}^{-\frac{1}{2}}$, where $\hat{\mathbf{A}} \in \mathbb{R}^{N\times N}$ denotes the self-looped adjacency matrix and $\hat{\mathbf{D}}$ is the degree matrix of $\hat{\mathbf{A}}$. Besides, $||\cdot||$ denotes the square norm. The notations are summarized in Table 1.

In this paper, we aim to group the nodes into several disjoint groups in the unsupervised manner. To be specific, we first embed the nodes into the latent space without labels and then directly performance clustering algorithm K-means [29] over the learned embeddings.

3.2 Graph Augmentation Module

In the field of self-supervised graph representation learning, several works [15], [32], [33] have demonstrated that, the network would learn richer node representations from different augmented graphs. Motivated by their success, we adopt the augmentations on graphs to improve the discriminative capability of node representations. As illustrated the graph augmentation module in Fig. 2, two types of augmentations are considered in our proposed method, i.e., feature perturbation and structure construction.

3.2.1 Feature Perturbation

First, we utilize an attribute-level augmentation, i.e., feature perturbation, which disturbs the attribute of nodes in the graph. To be specific, we firstly generate the random noise matrix $\mathbf{N} \in \mathbb{R}^{N\times D}$, which is sampled from a Gaussian distribution $\mathcal{N}(1, 0.1)$. Subsequently, we calculate the perturbated attribute matrix $\widetilde{\mathbf{X}} \in \mathbb{R}^{N\times D}$ as formulated:

$$\widetilde{\mathbf{X}} = \mathbf{X} \odot \mathbf{N}, \tag{1}$$

where $\odot$ denotes the Hadamard product [34].

Notations	Meanings
$\mathbf{X} \in \mathbb{R}^{N\times D}$	The attribute matrix
$\mathbf{A} \in \mathbb{R}^{N\times N}$	The original adjacency matrix
$\hat{\mathbf{A}} \in \mathbb{R}^{N\times N}$	The self-looped adjacency matrix
$\widetilde{\mathbf{A}} \in \mathbb{R}^{N\times N}$	The normalized adjacency matrix
$\mathbf{A}^f \in \mathbb{R}^{N\times N}$	The KNN graph adjacency matrix
$\mathbf{A}^d \in \mathbb{R}^{N\times N}$	The graph diffusion matrix
$\mathbf{Z}^{v_k} \in \mathbb{R}^{N\times d}$	The node embedding in k-th view
$\widetilde{\mathbf{Z}}^{v_k} \in \mathbb{R}^{d\times K}$	The cluster-level embedding in k-th view
$\mathbf{Z} \in \mathbb{R}^{N\times d}$	The clustering-oriented node embedding
$\mathbf{S}^{\mathcal{N}} \in \mathbb{R}^{N\times N}$	The cross-view sample correlation matrix
$\mathbf{S}^{\mathcal{F}} \in \mathbb{R}^{d\times d}$	The cross-view feature correlation matrix
$\mathbf{T} \in \mathbb{R}^{N\times N}$	The clustering-refined affinity matrix
$\mathbf{Q} \in \mathbb{R}^{N\times C}$	The soft assignment distribution
$\mathbf{P} \in \mathbb{R}^{N\times C}$	The target distribution

Table 1: Notation summary table

3.2.2 Structure Construction

Besides, for the structure-level augmentations, we consider two structure construction strategies. The first strategy is the KNN graph adjacency matrix construction based on the feature similarity [35]. Specifically, we calculate the cosine similarity between nodes in the graph and then generate the KNN graph adjacency matrix $\mathbf{A}^f \in \mathbb{R}^{N\times D}$ by the K-Nearest Neighbors algorithm (KNN) [36] as formulated:

$$\mathbf{A}^f = Softmax(KNN(\frac{\mathbf{X}\mathbf{X}^T}{||\mathbf{X}||||\mathbf{X}||})), \tag{2}$$

where $KNN(\cdot)$ denotes the KNN algorithm [36] and $Softmax(\cdot)$ denotes the softmax function [37] for normalization. For the number of nearest neighbors ϵ in the KNN algorithm [36], we set it to 5 in our model. To another strategy, we adopt the generalized graph diffusion utilized in MVGRL [15], which capture the local and global information of a graph structure. The generalized graph diffusion matrix $\mathbf{G} \in \mathbb{R}^{N\times N}$ could be formulated as:

$$\mathbf{G} = \sum_{k=0}^{\infty} \theta_k \mathbf{T}^k, \tag{3}$$

where $\theta_k \in [0, 1]$ is the coefficient k-order structure information and $\sum_{k=0}^{\infty} \theta_k = 1$. Besides, $\mathbf{T} \in \mathbb{R}^{N\times N}$ denotes the generalized transition matrix. Actually, we adopt a special case of the generalized graph diffusion, i.e., Personalized PageRank (PPR) algorithm [38], which sets $\mathbf{T} = \widetilde{\mathbf{A}}$ and $\theta_k = \alpha(1-\alpha)^k$. Thus, the graph diffusion matrix could be formulated by the closed-form solution to PPR as follow:

$$\mathbf{A}^d = \alpha(\mathbf{I}_N - (1-\alpha)\widetilde{\mathbf{A}})^{-1}, \tag{4}$$

where α is the teleport (or restart) probability and $\mathbf{I}_N \in \mathbb{R}^{N\times N}$ denotes an identity matrix.

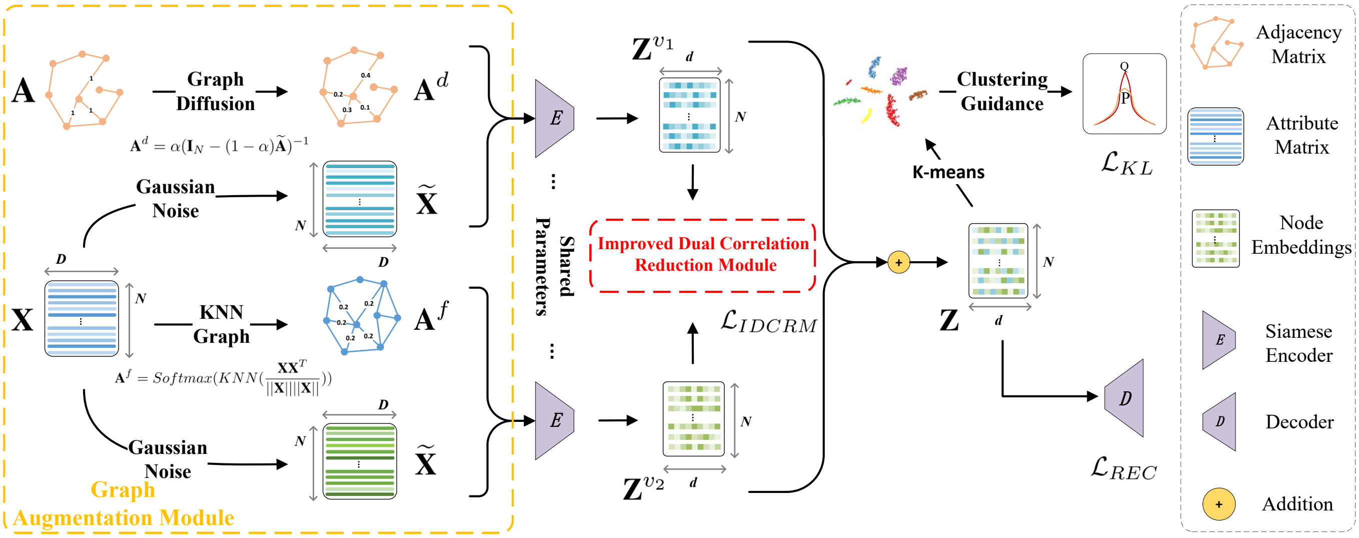


Figure 2: Illustration of the training process of the proposed IDCRN. In the graph augmentation module, two different minor Gaussian perturbations are added to the attribute matrix to generate two forms of the same matrix. The graph structure is strengthened with two manners, i.e., graph diffusion and KNN graph construction based on similarity to improve the graph quality. After the data augmentation, the generated attribute and graph structure pairs are embedded with a siamese network into the latent space. Then, by reducing the feature redundancy and correcting the embedded sample distribution with the improved dual correlation reduction module (IDCRM) in Fig. 3, we improve the discriminative capability of the network. Finally, the two embeddings are merged to perform sample reconstruction and K-means clustering [29], which is guided by the widely-used distribution alignment loss [2], [9], [11], [30], [31].

Based on the feature perturbation and the structure construction, two augmented graphs $\mathcal{G}^1 = \{\widetilde{\mathbf{X}}, \mathbf{A}^f\}$ and $\mathcal{G}^2 = \{\widetilde{\mathbf{X}}, \mathbf{A}^d\}$ are generated. In what follows, we aim to guide our network to learn the more discriminative embeddings from two augmented views of the graph.

3.3 Improved Dual Correlation Reduction Module

In this section, we propose Improved Dual Correlation Reduction Module (IDCRM) to improve the discriminative capability of the node embeddings in two aspects, i.e., the sample aspect and the feature aspect, thus avoiding the representation collapse problem. Following the above ideas, two strategies termed Affinity Recovery Strategy (ARS) and Redundancy Reduction Strategy (RRS) are designed in IDCRM as shown in Fig. 3.

3.3.1 Affinity Recovery Strategy

To the sample aspect, we design Affinity Recovery Strategy (ARS) to enhance the feature discriminative capability implicitly. Specifically, the proposed ARS contains the following three steps.

First, we encode two graphs $\mathcal{G}^1$ and $\mathcal{G}^2$, which are generated by the graph augmentation module, into two-view node embeddings $\mathbf{Z}^{v_1}$ and $\mathbf{Z}^{v_2}$ with a siamese graph encoder [11].

Second, the cross-view sample correlation matrix $\mathbf{S}^{\mathcal{N}} \in \mathbb{R}^{N \times N}$, whose elements comprised between -1 and 1, could be formulated as:

$$\mathbf{S}^{\mathcal{N}}_{ij} = \frac{(\mathbf{Z}^{v_1}_i)(\mathbf{Z}^{v_2}_j)^{\mathrm{T}}}{||\mathbf{Z}^{v_1}_i||||\mathbf{Z}^{v_2}_j||}, \ \forall\ i,j \in [1,N], \tag{5}$$

where the element $\mathbf{S}^{\mathcal{N}}_{ij}$ is the cosine similarity between $\mathbf{Z}^{v_1}_i$ and $\mathbf{Z}^{v_2}_j$. Besides, $\mathbf{Z}^{v_1}_i$ and $\mathbf{Z}^{v_2}_j$ denote i-th node embedding of the first view and j-th node embedding of the second view, respectively.

Subsequently, we force the cross-view sample correlation matrix $\mathbf{S}^{\mathcal{N}}$ to approximate the clustering-refined affinity matrix $\mathbf{T} \in \mathbb{R}^{N \times N}$ as formulated:

$$\begin{aligned}\mathcal{L}_N &= \frac{1}{N^2}\sum(\mathbf{S}^{\mathcal{N}} - \mathbf{T})^2 \\ &= \frac{1}{N^2}(\underbrace{\sum_i\sum_j \mathbb{1}^1_{ij}(\mathbf{S}^{\mathcal{N}}_{ij} - 1)^2}_{homogeneity} + \underbrace{\sum_i\sum_j \mathbb{1}^0_{ij}((\mathbf{S}^{\mathcal{N}}_{ij})^2)}_{heterogeneity}),\end{aligned} \tag{6}$$

where $\mathbb{1}^1_{ij}$ denotes if $\mathbf{T}_{ij}$ is equal to 1 while $\mathbb{1}^0_{ij}$ denotes if $\mathbf{T}_{ij}$ is equal to 0. We design $\mathbf{T}$ in two steps as shown in the left part of Fig. 3. (1) Considering the homogeneity principle [39], which indicates that nodes from the same class tend to form edges, we initialize $\mathbf{T}$ with $\hat{\mathbf{A}}$, i.e., the self-looped adjacency matrix. (2) $\mathbf{T}$ is refined with the 60% high confident clustering resultant samples. To be specific, we construct pseudo labels for these samples based on the cluster-ID and further add / remove edges when the paired samples have the same / different pseudo labels. It's worth mentioning that the samples, which are closer to the corresponding cluster centers, have higher confidence in K-means clustering algorithm [29]. In this manner, the proposed clustering-refined affinity matrix $\mathbf{T}$ could better reveal the homogeneity between the nodes from the same categories and the heterogeneity between the nodes from the different categories.

In Eq. (6), the *homogeneity* term pulls together the nodes from the same category across two views. Differently, the *heterogeneity* term pushes away the samples from different categories across two views. In this manner, our proposed ARS guides the learned representation to recover the affinity matrix even across views, thus improving the feature discriminative capability implicitly.

3.3.2 Redundancy Reduction Strategy

In addition to the sample aspect, we further consider the feature aspect to reduce the redundancy between different dimensions of

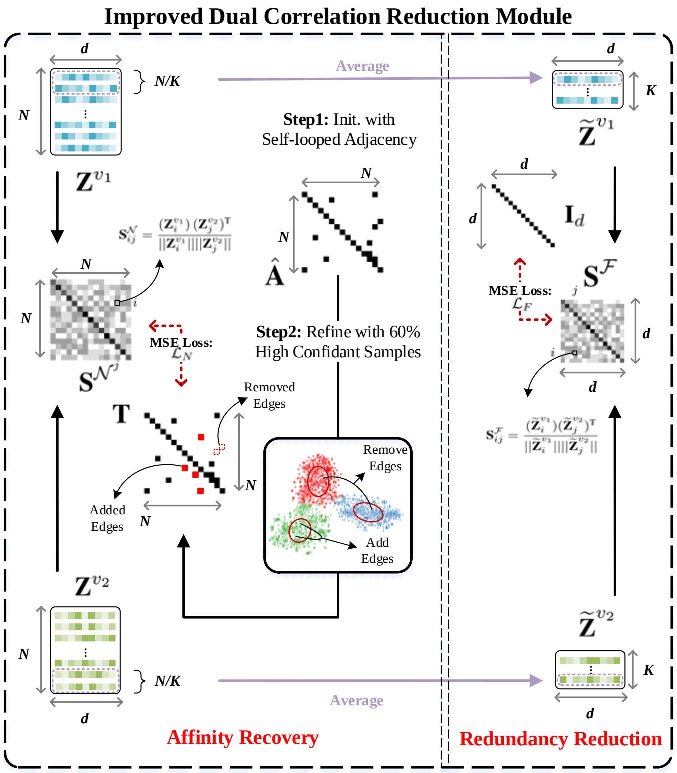

Improved Dual Correlation Reduction Module
d
N/K
N
Average
K
$\widetilde{\mathbf{Z}}^{v_1}$
$\mathbf{Z}^{v_1}$
Step1: Init. with Self-looped Adjacency
N
$\hat{\mathbf{A}}$
d
$\mathbf{I}_d$
$\mathbf{S}^{\mathcal{N}}_{ij} = \frac{(\mathbf{Z}^{v_1}_i)(\mathbf{Z}^{v_2}_j)^{\mathrm{T}}}{||\mathbf{Z}^{v_1}_i||||\mathbf{Z}^{v_2}_j||}$
MSE Loss: $\mathcal{L}_F$
$\mathbf{S}^{\mathcal{F}}$
$\mathbf{S}^{\mathcal{N}}$
MSE Loss: $\mathcal{L}_N$
Step2: Refine with 60% High Confidant Samples
T
Removed Edges
Remove Edges
Added Edges
Add Edges
$\mathbf{S}^{\mathcal{F}}_{ij} = \frac{(\widetilde{\mathbf{Z}}^{v_1}_i)(\widetilde{\mathbf{Z}}^{v_2}_j)^{\mathrm{T}}}{||\widetilde{\mathbf{Z}}^{v_1}_i||||\widetilde{\mathbf{Z}}^{v_2}_j||}$
$\mathbf{Z}^{v_2}$
$\widetilde{\mathbf{Z}}^{v_2}$
Average
Affinity Recovery
Redundancy Reduction

Figure 3: Illustration of the Improved Dual Correlation Reduction Module (IDCRM). Our proposed IDCRM aims to improve the discriminative capability of the embeddings in two aspects, i.e., the sample aspect and the feature aspect. Specifically, to the feature aspect, we reduce the redundancy between different dimensions of features via approximating the cross-view feature correlation matrix to the identity matrix, thus enhancing the discriminative capability of the latent space explicitly. Moreover, in the sample aspect, we force the cross-view sample correlation matrix to approximate the high confident clustering results refined affinity matrix. With this setting, we guide the learned latent representation to recover the affinity matrix even across views, thus improving the feature discriminative capability implicitly.

the latent features. Guided by this idea, anther effective strategy termed Redundancy Reduction Strategy (RRS) is designed as illustrated in the right part of Fig. 3. To be specific, our proposed RRS contains the following three steps.

We first utilize a readout function $\mathcal{R}(\cdot) : \mathbb{R}^{d\times N} \rightarrow \mathbb{R}^{d\times K}$ to obtain the cluster-level embeddings $\widetilde{\mathbf{Z}}^{v_1}, \widetilde{\mathbf{Z}}^{v_2} \in \mathbb{R}^{d\times K}$ from the node embeddings $\mathbf{Z}^{v_1}, \mathbf{Z}^{v_2}$. Here, to the readout function $\mathcal{R}$, we first divide the samples into K groups and then output the average value of each group.

Second, similar to Eq. (5), we calculate the cross-view feature correlation matrix $\mathbf{S}^{\mathcal{F}} \in \mathbb{R}^{N\times N}$ as formulated:

$$\mathbf{S}^{\mathcal{F}}_{ij} = \frac{(\widetilde{\mathbf{Z}}^{v_1}_i)(\widetilde{\mathbf{Z}}^{v_2}_j)^{\mathrm{T}}}{||\widetilde{\mathbf{Z}}^{v_1}_i||||\widetilde{\mathbf{Z}}^{v_2}_j||},\ \forall\ i,j \in [1,d], \tag{7}$$

where $\mathbf{S}^{\mathcal{F}}_{ij}$ actually denotes the cosine similarity between i-th dimension feature in the first view and j-th dimension feature in the second view.

Subsequently, different from Eq. (6), we force the cross-view feature correlation matrix $\mathbf{S}^{\mathcal{F}}$ to approximate an identity matrix $\mathbf{I}_d \in \mathbb{R}^{d\times d}$ as formulated:

$$\begin{aligned}\mathcal{L}_F &= \frac{1}{d^2}\sum(\mathbf{S}^{\mathcal{F}} - \mathbf{I}_d)^2 \\ &= \frac{1}{d^2}\sum_{i=1}^{d}\left(\mathbf{S}^{\mathcal{F}}_{ii} - 1\right)^2 + \frac{1}{d^2-d}\sum_{i=1}^{d}\sum_{j\neq i}\left(\mathbf{S}^{\mathcal{F}}_{ij}\right)^2,\end{aligned} \tag{8}$$

where d denotes the dimension of learned embedding.

We analyze Eq. (8) that the first term indicates that the same dimensions of the learned features from two augmented views are enforced to agree with each other. On the contrary, the second term decorrelates the different dimensions of the latent representations. By this way, the redundant information in the learned features is reduced and then the discriminative capability of features is enhanced explicitly, thus avoiding the representation collapse problem.

During training process of the network, we adopt a propagation regularization [17] to alleviate over-smoothing as formulated:

$$\mathcal{L}_R = JSD(\mathbf{Z},\ \widetilde{\mathbf{A}}\mathbf{Z}), \tag{9}$$

where $JSD(\cdot)$ is the Jensen-Shannon divergence [40]. In this manner, IDCRN is enabled to capture long-range information with the shallow network.

3.3.3 Fusion and Clustering

Under the constraints of ARS and RRS, we combination the two views of the node embeddings in a linear manner as formulated:

$$\mathbf{Z} = \frac{1}{2}(\mathbf{Z}^{v_1} + \mathbf{Z}^{v_2}), \tag{10}$$

where $\mathbf{Z} \in \mathbb{R}^{N\times d}$ denotes the resultant clustering-oriented node embeddings. Subsequently, we directly perform K-means algorithm [29] over $\mathbf{Z}$ and obtain the clustering results.

In summary, the loss function of the proposed IDCRM could be formulated as follows:

$$\mathcal{L}_{IDCRM} = \mathcal{L}_N + \mathcal{L}_F + \gamma\mathcal{L}_R, \tag{11}$$

where γ is a trade-off hyper-parameter. Technically, in the proposed IDCRM, we consider to enhance the discriminative capability of the node embeddings from both the sample and feature perspective. Under the constraint of IDCRM, our network is guided to reveal the underlying sample distribution and meanwhile the redundancy in the learned features could be filtered out. In this manner, our model would learn more discriminative embeddings to avoid the collapsed representation and further improve the clustering performance.

3.4 Overall Objective Function

The overall objective function of IDCRN contains three parts, i.e., the reconstruction loss, the clustering loss, and the loss of IDCRM as follows:

$$\mathcal{L} = \mathcal{L}_{IDCRM} + \mathcal{L}_{REC} + \lambda\mathcal{L}_{KL}, \tag{12}$$

where $\mathcal{L}_{REC}$ is the MSE reconstruction loss adopted in [11]. And $\mathcal{L}_{KL}$ denotes the KL divergence [41], i.e., a widely-used clustering loss in [2], [9], [11], [30], [31]. Here, we first generate a soft assignment distribution $\mathbf{Q} \in \mathbb{R}^{N\times C}$ and a target distribution $\mathbf{P} \in \mathbb{R}^{N\times C}$ over the node embeddings $\mathbf{Z}$. And then we align them by $\mathcal{L}_{KL}$ to guide the network. The detailed procedure of IDCRN is shown in Algorithm 1.

Algorithm 1 IDCRN

Input: An undircted graph: $\mathcal{G} = \{\mathbf{X}, \mathbf{A}\}$; The cluster number C; Iteration number t; Hyper-parameters γ and λ.
Output: The clustering result $\mathbf{O}$.

```
Utilize the proposed graph augmentation module to generate two
  augmented graph views $\mathcal{G}^1 = \{\widetilde{\mathbf{X}}, \mathbf{A}^f\}$ and $\mathcal{G}^2 = \{\widetilde{\mathbf{X}}, \mathbf{A}^d\}$;
Pre-train the feature extraction encoder to obtain $\mathbf{Z}$;
Initialize the cluster centers by performing K-means over $\mathbf{Z}$;
for $i = 1$ to $t$ do
    Utilize the feature extraction encoder to obtain $\mathbf{Z}^{v_1}$ and $\mathbf{Z}^{v_2}$;
    Obtain $\widetilde{\mathbf{Z}}^{v_1}$ and $\widetilde{\mathbf{Z}}^{v_2}$ by the readout function $\mathcal{R}$;
    Calculate $\mathbf{S}^{\mathcal{N}}$ and $\mathbf{S}^{\mathcal{F}}$ by Eq. (5) and Eq. (7), respectively;
    Conduct the Affinity Recovery Strategy and Redundancy Reduction Strategy by Eq. 6 and Eq. 8, respectively;
    Obtain $\mathbf{Z}$ by fusing $\mathbf{Z}^{v_1}$ and $\mathbf{Z}^{v_2}$ in Eq. (10);
    Calculate $\mathcal{L}_{IDCRM}$, $\mathcal{L}_{REC}$, and $\mathcal{L}_{KL}$, respectively.
    Optimize the whole network by minimizing $\mathcal{L}$ in Eq. (12);
end for
Obtain $\mathbf{O}$ by performing K-means over $\mathbf{Z}$.
return $\mathbf{O}$
```

4 EXPERIMENTS

4.1 Datasets

To verify the effectiveness and efficiency of IDCRN, abundant experimental studies are conducted on six graph clustering benchmarks, including ACM [9], CITE [9], DBLP [9], AMAP [42], PUBMED [43], and CORAFULL [44]. We list the statistics of these datasets in Table 2 and the detailed descriptions are summarized as follows:

- **ACM** [9]: It is a network of the papers. An edge will be constructed between two papers if they are written by the same author. The features of the papers are the bag-of-words of the keywords. The papers published in MobiCOMM, SIGCOMM, SIGMOD, KDD are selected and divided into three classes, including data mining, wireless communication, and database.
- **CITE** [9]: This citation network consists of a set of citation links between different documents whose feature vectors are the sparse bag-of-words. The labels are divided into the six areas including HCI, machine language, information retrieval, database, artificial intelligence, and agents.
- **DBLP** [9]: This author network contains authors from four areas including information retrieval, machine learning, data mining, and database. The edge in constructed between two authors if they are the co-author relationship. The features of the authors are the bag-of-words of keywords.
- **AMAP** [42]: This is a co-purchase graph from Amazon. The nodes in the graph denote the products and the features are the reviews encoded by the bag-of-words. The edges indicate whether two products are frequently co-purchased or not. The nodes are divided into eight classes.
- **PUBMED** [43]: This is a citation network, which contains scientific publications from the PubMed database. The nodes are divided into three classes and links indicates the citation between different publications. The publications in the graph are described by a TF/IDF weighted word vector from a dictionary which consists of 500 unique words.
- **CORAFULL** [44]: The is a citation network consists of 19793 scientific publications classified into one of seventy classes. This citation network includes 65311 links.

Dataset	Samples	Dimensions	Classes
DBLP	4057	334	4
CITE	3327	3703	6
ACM	3025	1870	3
AMAP	7650	745	8
PUBMED	19717	500	3
CORAFULL	19793	8710	70

Table 2: Dataset summary

4.2 Experiment Setup

4.2.1 Training Procedure

The deep learning platform and the GPU of all experiments are PyTorch and an NVIDIA 3090. The training process of our network consists of three steps. Following DFCN [11], we independently pre-train the sub-networks for 30 epochs by minimizing the reconstruction loss $\mathcal{L}_{REC}$. Afterward, we obtain the initial clustering centers by integrating two sub-networks into a united framework and training another 100 epochs. Then we fine-tune our whole network with 400 epochs until convergence by minimizing the loss calculated in Eq. 12. Consequently, we perform clustering on the embeddings $\mathbf{Z}$ by K-means algorithm [29]. In the compare experiments, we conduct ten runs for all methods and report the average values with standard deviations of four metrics to alleviate the random seed influence.

4.2.2 Parameters Setting

To ARGA / ARVGA [3], MVGRL [15], and DFCN [11], we reproduce the average values with standard deviations as results by adopting the corresponding source code with the original literature setting. To MCGC [20], for fairness, we merely adopt and run their source code on the graph datasets in Table 2. For other baselines, we list the corresponding results reported in DFCN [11]. In our proposed method, we utilize DFCN [11] as our feature extractor. Besides, our network is optimized with the Adam optimizer [45]. The learning rate of our IDCRN is set to 5e-5 for ACM, 1e-3 for AMAP, 1e-4 for DBLP, 1e-5 for CITE, PUBMED, and CORAFULL, respectively. The teleport probability α in the Personalized PageRank (PPR) algorithm [38] is set as 0.1 for PUBMED, 0.3 for ACM, and 0.2 for other datasets. Afterward, ϵ in KNN algorithm [36] and K in readout function $\mathcal{R}(\cdot)$ are fixed as 5 and the number of clusters C. For the trade-off hyper-parameters λ and γ, we respectively set them as 10 and 1e3.

4.2.3 Metrics

To comprehensively verify the superiority of the compared methods, the clustering performance is evaluated by four metrics [46], [47], [48], [49], [50], i.e., ACC, NMI, ARI, and F1. In more detail, ACC could be calculated as follow:

$$\mathrm{ACC} = \frac{\sum_{i=1}^{n} \phi(l_i, map(c_i))}{n}, \tag{13}$$

where c_i and l_i respectively denote the predicted cluster ID and the label for the i-th sample. The$\phi(\cdot)$ is an indicator function as formulated:

$$\phi(l_i, map(c_i)) = \begin{cases} 1 \;\; if \; l_i = map(c_i), \\ 0 \;\; otherwise. \end{cases} \tag{14}$$

The best map from the predicted cluster ID c_i to class ID could be constructed by the Kuhn-Munkres algorithm [51], i.e., the $map(\cdot)$ function.

Another critical metric macro F1-score, which indicates a balance of precision and recall, cloud be calculated as formulated:

$$\mathrm{F1} = 2 \cdot \frac{\mathrm{P} \cdot \mathrm{R}}{\mathrm{P} + \mathrm{R}}. \tag{15}$$

In detail, $\mathrm{P} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FP})$ is the precision value and $\mathrm{R} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FN})$ is the recall value. TP, FP, and FN denotes True Positive error, False Positive error, and False Negative error, respectively.

A mutual information score based metric named NMI is widely used in clustering tasks since it is robust to the unbalanced label distribution. It is defined as:

$$\mathrm{NMI} = -\frac{2\sum_x \sum_y p(x,y) log\frac{p(x,y)}{p(x)p(y)}}{\sum_i p(x_i) log(p(x_i)) + \sum_j p(y_j) log(p(y_j))}, \tag{16}$$

where x, y denote the distribution of the predicted results and the ground truth, respectively.

Different from NMI, ARI is based on the similarity of pairwise labels between the ground truth and predicted results as formulated:

$$\mathrm{ARI} = \frac{\overbrace{\sum_{ij}\binom{n_{ij}}{2}}^{\text{Index}} - \overbrace{\left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{n}{2}}^{\text{Expected index}}}{\underbrace{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right]}_{\text{Max index}} - \underbrace{\left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{n}{2}}_{\text{Expected index}}}, \tag{17}$$

where n is the number of all pairs, a is the number of pairs with the same cluster, and b is the number of pairs with different clusters.

4.3 Performance Comparison

In this section, we conduct comparison experiments of IDCRN and the other 14 baselines to show the superiority of IDCRN. Specifically, K-means algorithm [29] is a classic clustering method with the idea of EM algorithm [53]. Besides, three generative deep clustering methods, including AE [19], DEC [30], and IDEC [31], first train an auto encoder to embed the samples into the latent space and then perform K-means [29] over the learned embeddings. Different from them, three typical GCN-based frameworks, i.e., GAE / VGAE [14], DAEGC [2], and ARGA / ARVGA [3] aim to learn representations for clustering by exploiting from both the structure and attribute of the graph. More recently, four state-of-the-art deep graph clustering methods, i.e., SDCN / SDCN$_Q$ [9], DFCN [11], MVGRL [15], and MCGC [20] have achieved the promising clustering performance through learning the consensus representations from different views of the graph.

Table 3 reports the clustering performance of IDCRN and other 14 compared baselines on six benchmarks. Based on these results, we analyze and conclude as follows.

- Our proposed IDCRN almost exceeds all other baselines in four metrics on six datasets except the NMI metric on PUBMED dataset.
- Specifically, our proposed method achieves better clustering performance than the strongest deep graph clustering frameworks, including SDCN/SDCN$_Q$ [9], MVGRL [15], MCGC [20], and DFCN [11]. For instance, IDCRN exceeds DFCN by 6.08% 9.00%, 11.81% 5.77% increment concerning ACC, NMI, ARI, and F1 metrics on DBLP dataset. The reason is that they all aim to learn latent embeddings from multi-view graph data with redundant information, thus easily suffering from representation collapse. Different from them, by reducing the redundancy and recovering the affinity matrix, IDCRN is guided to learn more discriminative representation, thus avoiding the collapsed representation.
- Other GCN-based graph clustering methods, including ARGA [3], DAEGC [2], and GAE/VGAE [14] achieve unsatisfactory performance compared to ours since these methods fail to exploit different views of the graph.
- The auto-encoder-based clustering methods, including AE [19], DEC [19], and IDEC [31], achieve unpromising clustering performance. This verifies that these methods, which are merely based on attribute of the samples, can not learn discriminative features from the graph data.
- The classical clustering method K-means algorithm [29] achieve unpromising results since it is directly performed on the raw attributes.

Overall, the above observations and conclusions have verified that our proposed method effectively alleviates representation collapse and achieves superior clustering performance.

4.4 Ablation Studies

In this section, we conduct ablation studies to verify the effectiveness of our proposed IDCRM and further two proposed strategies including Affinity Recovery Strategy (ARS) and Redundancy Reduction Strategy (RRS) in IDCRM.

4.4.1 Effectiveness of Improved Dual Correlation Reduction Module

In order to verify the effectiveness of Improved Dual Correlation Reduction Module (IDCRM) clearly, extensive ablation studies are conducted in Table 4. Here, we adopt DFCN [11] as the baseline. Besides, the baseline with the propagated regularization (P-reg) [17], the IDCRM, and both of them is denoted as B-P, B-I, and, B-P-I, respectively. From these results, we could observe and conclude as follows.

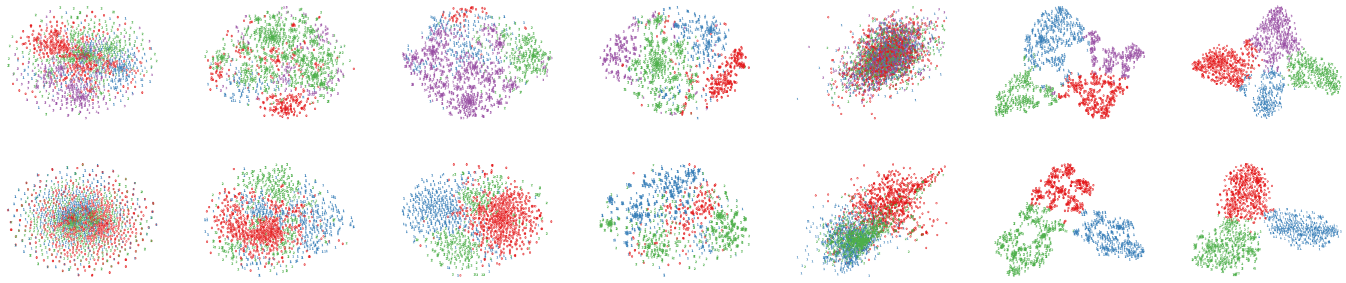
- To B-P, P-reg could improve the baseline by about 0.79% on DBLP dataset. From these results, we conclude that P-reg could some extent alleviate the over-smoothing problem and improve our model's generalization capacity.
- Our proposed IDCRM improves the baseline by a large margin. For instance, the baseline with IDCRM, i.e., B-I, exceeds the baseline by 6.04%, 9.02%, 11.74%, 5.70% performance increment in terms of ACC, NMI, ARI, and F1 on DBLP dataset. Based on these results, we analyze and conclude that the discriminative capacity of the latent features is enhanced by our proposed IDCRM, thus improving the clustering performance.
- Compared to other variants, B-P-I achieves the best results, verifying the effectiveness of the both components, i.e., IDCRM and P-reg.

4.4.2 Effectiveness of ARS and RRS

Furthermore, we carry on the ablation studies of the two proposed strategy including Affinity Recovery Strategy (ARS) and Redundancy Reduction Strategy (RRS). Here, we adopt DFCN [11] as our baseline. Then we denote B-R, B-A and, B-R-A as the baseline

Dataset	Metric	K-Means	AE	DEC	IDEC	GAE	VGAE	DAEGC	ARGA	ARVGA	SDCN$_Q$	SDCN	MVGRL	MCGC	DFCN	DCRN	IDCRN
		[29]	[19]	[30]	[31]	[14]	[14]	[2]	[3]	[3]	[9]	[9]	[15]	[20]	[11]	Our Proposed Methods	
DBLP	ACC	38.65±0.65	51.43±0.35	58.16±0.56	60.31±0.62	61.21±1.22	58.59±0.06	62.05±0.48	64.83±0.59	54.41±0.42	65.74±1.34	68.05±1.81	42.73±1.02	58.92±0.05	76.00±0.80	79.66±0.25	82.08±0.18
	NMI	11.45±0.38	25.40±0.16	29.51±0.28	31.17±0.50	30.80±0.91	26.92±0.06	32.49±0.45	29.42±0.92	25.90±0.33	35.11±1.05	39.50±1.34	15.41±0.63	33.69±0.06	43.70±1.00	48.95±0.44	52.70±0.36
	ARI	6.97±0.39	12.21±0.43	23.92±0.39	25.37±0.60	22.02±1.40	17.92±0.07	21.03±0.52	27.99±0.91	19.81±0.42	34.00±1.76	39.15±2.01	8.22±0.50	25.97±0.21	47.00±1.50	53.60±0.46	58.81±0.37
	F1	31.92±0.27	52.53±0.36	59.38±0.51	61.33±0.56	61.41±2.23	58.69±0.07	61.75±0.67	64.97±0.66	55.37±0.40	65.78±1.22	67.71±1.51	40.52±1.51	50.39±0.09	75.70±0.80	79.28±0.26	81.47±0.20
CITE	ACC	39.32±3.17	57.08±0.13	55.89±0.20	60.49±1.42	61.35±0.80	60.97±0.36	64.54±1.39	61.07±0.49	59.31±1.38	61.67±1.05	65.96±0.31	68.66±0.36	64.76±0.07	69.50±0.20	70.86±0.18	71.40±0.08
	NMI	16.94±3.22	27.64±0.08	28.34±0.30	27.17±2.40	34.63±0.65	32.69±0.27	36.41±0.86	34.40±0.71	31.80±0.81	34.39±1.22	38.71±0.32	43.66±0.40	39.11±0.06	43.90±0.20	45.86±0.35	46.77±0.21
	ARI	13.43±3.02	29.31±0.14	28.12±0.36	25.70±2.65	33.55±1.18	33.13±0.53	37.78±1.24	34.32±0.70	31.28±1.22	35.50±1.49	40.17±0.43	44.27±0.73	37.54±0.12	45.50±0.30	47.64±0.30	48.67±0.20
	F1	36.08±3.53	53.80±0.11	52.62±0.17	61.62±1.39	57.36±0.82	57.70±0.49	62.20±1.32	58.23±0.31	56.05±1.13	57.82±0.98	63.62±0.24	63.71±0.39	59.64±0.05	64.30±0.20	65.83±0.21	66.27±0.21
ACM	ACC	67.31±0.71	81.83±0.08	84.33±0.76	85.12±0.52	84.52±1.44	84.13±0.22	86.94±2.83	86.29±0.36	83.89±0.54	86.95±0.08	90.45±0.18	86.73±0.76	91.64±0.00	90.90±0.20	91.93±0.20	92.58±0.08
	NMI	32.44±0.46	49.30±0.16	54.54±1.51	56.61±1.16	55.38±1.92	53.20±0.52	56.18±4.15	56.21±0.82	51.88±1.04	58.90±0.17	68.31±0.25	60.87±1.40	70.71±0.00	69.40±0.40	71.56±0.61	73.17±0.32
	ARI	30.60±0.69	54.64±0.16	60.64±1.87	62.16±1.50	59.46±3.10	57.72±0.67	59.35±3.89	63.37±0.86	57.77±1.17	65.25±0.19	73.91±0.40	65.07±1.76	76.63±0.00	74.90±0.40	77.56±0.52	79.18±0.22
	F1	67.57±0.74	82.01±0.08	84.51±0.74	85.11±0.48	84.65±1.33	84.17±0.23	87.07±2.79	86.31±0.35	83.87±0.55	86.84±0.09	90.42±0.19	86.85±0.72	91.70±0.00	90.80±0.20	91.94±0.20	92.60±0.08
AMAP	ACC	27.22±0.76	48.25±0.08	47.22±0.08	47.62±0.08	71.57±2.48	74.26±3.63	76.44±0.01	69.28±2.30	61.46±2.71	35.53±0.39	53.44±0.81	45.19±1.79	71.64±0.00	76.88±0.80	79.94±0.13	80.17±0.04
	NMI	13.23±1.33	38.76±0.30	37.35±0.05	37.83±0.08	62.13±2.79	66.01±3.40	65.57±0.03	58.36±2.76	53.25±1.91	27.90±0.40	44.85±0.83	36.89±1.31	61.54±0.00	69.21±1.00	73.70±0.24	74.32±0.07
	ARI	5.50±0.44	20.80±0.47	18.59±0.04	19.24±0.07	48.82±4.57	56.24±4.66	59.39±0.02	44.18±4.41	38.44±4.69	15.27±0.37	31.21±1.23	18.79±0.47	43.23±0.00	58.98±0.84	63.69±0.20	64.10±0.10
	F1	23.96±0.51	47.87±0.20	46.71±0.12	47.20±0.11	68.08±1.76	70.38±2.98	69.97±0.02	64.30±1.95	58.5±1.70	34.25±0.44	50.66±1.49	39.65±2.39	68.64±0.00	71.58±0.31	73.82±0.12	74.01±0.04
PUBMED	ACC	59.83±0.01	63.07±0.31	60.14±0.09	60.70±0.34	62.09±0.81	68.48±0.77	68.73±0.03	65.26±0.12	64.25±1.24	64.39±0.30	64.20±1.30	67.01±0.52	60.97±0.01	68.89±0.07	69.87±0.07	70.02±0.03
	NMI	31.05±0.02	26.32±0.57	22.44±0.14	23.67±0.29	23.84±3.54	30.61±1.71	28.26±0.03	24.80±0.17	23.88±1.05	26.67±1.31	22.87±2.04	31.59±1.45	33.39±0.02	31.43±0.13	32.20±0.08	33.29±0.07
	ARI	51.43±0.35	23.86±0.67	19.55±0.13	20.58±0.39	20.62±1.39	30.15±1.23	29.84±0.04	24.35±0.17	22.82±1.52	24.61±1.46	22.30±2.07	29.42±1.06	29.25±0.01	30.64±0.11	31.41±0.12	32.67±0.05
	F1	58.88±0.01	64.01±0.29	61.49±0.10	62.41±0.32	61.37±0.85	67.68±0.89	68.23±0.02	65.69±0.13	64.51±1.32	65.46±0.39	65.01±1.21	67.07±0.36	59.84±0.01	68.10±0.07	68.94±0.08	69.19±0.03
CORAFULL	ACC	26.27±1.10	33.12±0.19	31.92±0.45	32.19±0.31	29.60±0.81	32.66±1.29	34.35±1.00	22.07±0.43	29.57±0.59	29.75±0.69	26.67±0.40	31.52±2.95	29.08±0.58	37.51±0.81	38.80±0.60	39.45±0.50
	NMI	34.68±0.84	41.53±0.25	41.67±0.24	41.64±0.28	45.82±0.75	47.38±1.59	49.16±0.73	41.28±0.25	48.77±0.44	40.10±0.22	37.38±0.39	48.99±3.95	36.86±0.56	51.30±0.41	51.91±0.35	52.83±0.39
	ARI	9.35±0.57	18.13±0.27	16.98±0.29	17.17±0.22	17.84±0.86	20.01±1.38	22.60±0.47	12.38±0.24	18.80±0.57	16.47±0.38	13.63±0.27	19.11±2.63	13.15±0.48	24.46±0.48	25.25±0.49	25.97±0.54
	F1	22.57±1.09	28.4±0.30	27.71±0.58	27.72±0.41	25.95±0.75	29.06±1.15	26.96±1.33	18.85±0.41	25.43±0.62	24.62±0.53	22.14±0.43	26.51±2.87	22.90±0.52	31.22±0.87	31.68±0.76	32.58±0.72

Table 3: The clustering performance of 14 state-of-the-art algorithms and our proposed method with mean values ± standard deviations (mean ± std) on six datasets.The values with red and blue correspond to the best and the runner-up results.



(a) Raw Data (b) AE (c) DEC (d) GAE (e) ARGA (f) DFCN (g) IDCRN

Figure 4: t-SNE [52] visualization of the representation of raw data, AE [19], DEC [30], GAE [14], ARGA [3], DFCN [11], and our proposed method on two datasets. The first and second row indicate the results on ACM and DBLP dataset, respectively.

Dataset	Metric	B	B-P	B-I	B-P-I
DBLP	ACC	76.00±0.80	77.00±0.41	82.04±0.22	82.08±0.18
	NMI	43.70±1.00	44.98±0.56	52.72±0.42	52.70±0.36
	ARI	47.00±1.50	48.51±0.84	58.74±0.42	58.81±0.37
	F1	75.70±0.80	76.77±0.38	81.40±0.24	81.47±0.20
CITE	ACC	69.50±0.20	70.07±0.21	71.12±0.14	71.40±0.08
	NMI	43.90±0.20	44.75±0.40	46.30±0.35	46.77±0.21
	ARI	45.50±0.30	46.52±0.36	48.29±0.32	48.67±0.20
	F1	64.30±0.20	65.03±0.23	66.23±0.22	66.27±0.21
ACM	ACC	90.90±0.20	91.57±0.12	92.30±0.19	92.58±0.08
	NMI	69.40±0.40	70.82±0.25	72.32±0.53	73.17±0.32
	ARI	74.90±0.40	76.68±0.28	78.46±0.49	79.18±0.22
	F1	90.80±0.20	91.53±0.12	92.31±0.20	92.60±0.08
AMAP	ACC	76.88±0.80	79.01±0.01	80.02±0.24	80.17±0.04
	NMI	69.21±1.00	72.29±0.01	73.81±0.21	74.32±0.07
	ARI	58.98±0.84	62.10±0.01	63.95±0.39	64.10±0.10
	F1	71.58±0.31	73.09±0.00	73.92±0.20	74.01±0.04
PUBMED	ACC	68.89±0.07	69.43±0.05	69.80±0.06	70.02±0.03
	NMI	31.43±0.13	31.98±0.12	32.05±0.06	33.29±0.07
	ARI	30.64±0.11	31.35±0.12	31.34±0.11	32.67±0.05
	F1	68.10±0.07	68.54±0.06	68.83±0.07	69.19±0.03
CORAFULL	ACC	37.51±0.81	37.04±0.71	38.45±0.27	39.45±0.50
	NMI	51.30±0.41	51.90±0.26	51.04±0.23	52.83±0.39
	ARI	24.46±0.48	24.13±0.51	24.96±0.20	25.97±0.54
	F1	31.22±0.87	30.35±0.87	31.87±0.75	32.58±0.72

Table 4: Ablation study results of IDCRM and the propagated regularization on six benchmarks.

with RRS, ARS, and both, respectively. By observing the results in Fig. 5, we have three following three conclusions.

- B-R achieves better clustering performance than the baseline on four of six datasets since the learned embeddings is not robust without revealing the underlying sample distribution.
- The baseline with ARS significantly outperforms the baseline on six datasets. Take the results on DBLP dataset for an instance, B-A obtains 5.63% accuracy improvement. Benefited from our proposed ARS, the network is guided to reveal the underlying sample distribution, thus enhancing the discriminative capability of the learned features.
- Moreover, B-R-A achieves the best clustering performance, further indicating that our proposed RRS and ARS effectively improve the discriminative capability of the learned embeddings.

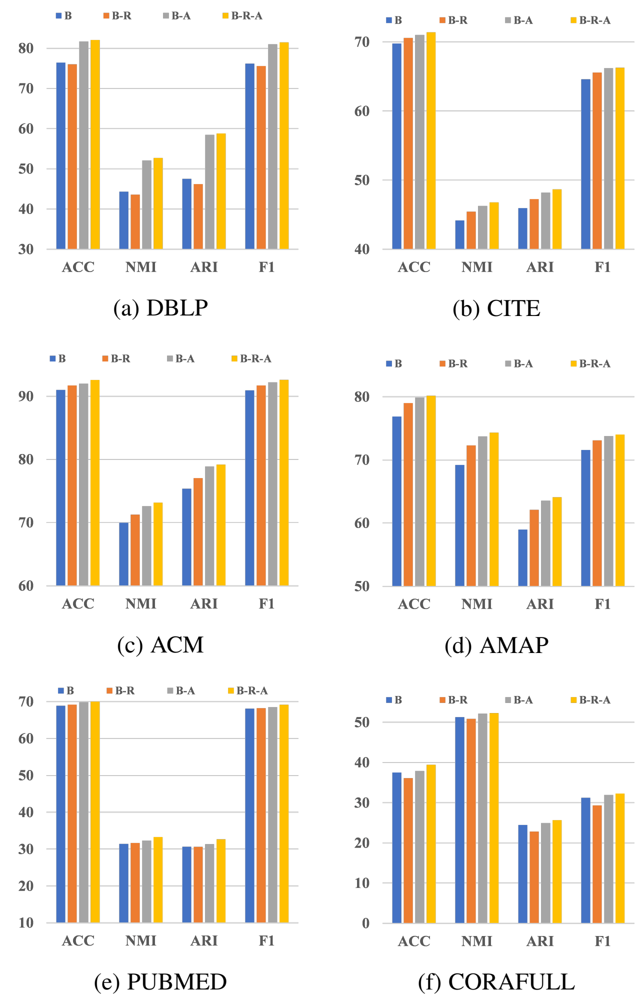
4.5 Sensitivity Analysis of Hyper-parameters

We conduct extensive experiments to analyze the robustness of our proposed method IDCRN to the hyper-parameters.

4.5.1 Sensitivity Analysis of Graph Augmentation Module

In order to investigate the influence of graph augmentation module, we first conduct an experiment about the teleport probability α in the graph diffusion, i.e., Personalized PageRank (PPR) [38], on DBLP, CITE, ACM, and AMAP datasets. In Fig. 6, we could observe that the accuracy firstly raises and reaches the high peak value where the teleport probability α is around 0.2 while the clustering performance decreases down with the larger teleport probability α. Besides, our proposed method is robust to the teleport probability α when $\alpha \in (0.2, 0.8)$.

In addition, we also explore the influence of the hyper-parameter ϵ in K-nearest neighbors algorithm (KNN) [36] during

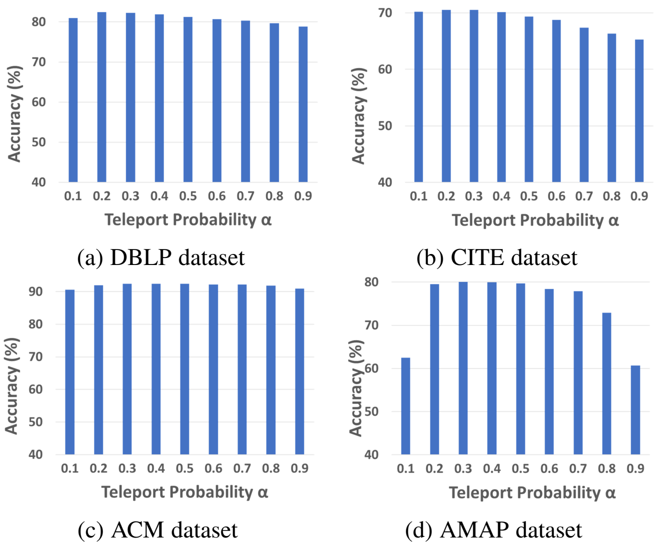


(a) DBLP (b) CITE

(c) ACM (d) AMAP

(e) PUBMED (f) CORAFULL

Figure 5: The ablation study results of the proposed two strategies, i.e., Affinity Recovery Strategy (ARS) and Redundancy Reduction Strategy (RRS).

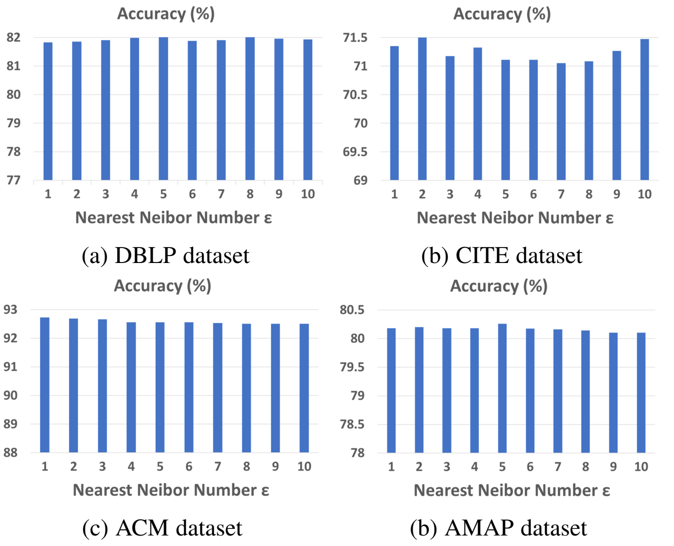


(a) DBLP dataset (b) CITE dataset

(c) ACM dataset (d) AMAP dataset

Figure 6: Sensitivity Analysis of the teleport probability α in the graph diffusion on DBLP, CITE, ACM, and AMAP datasets.

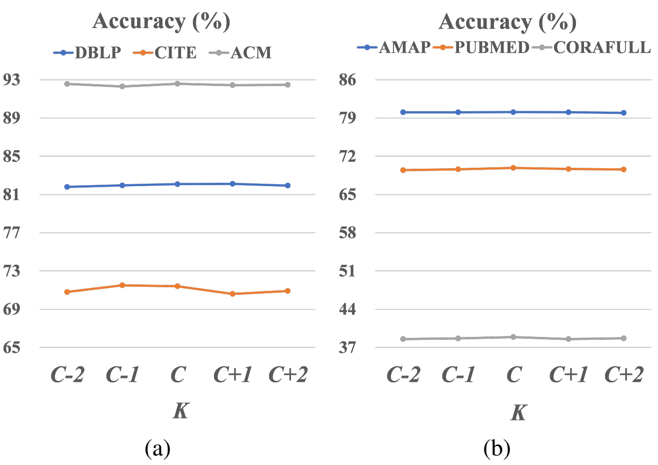
the process of the KNN graph adjacency matrix generation. From Fig. 7, we observe and conclude the ACC metric of clustering does not fluctuate significantly with the variation of ϵ so that our proposed IDCRN is insensitive to the nearest neighbor number ϵ.



(a) DBLP dataset (b) CITE dataset

(c) ACM dataset (b) AMAP dataset

Figure 7: Sensitivity Analysis of the nearest neighbor number ϵ in the KNN adjacency matrix generation on DBLP, CITE, ACM and AMAP datasets.



(a) (b)

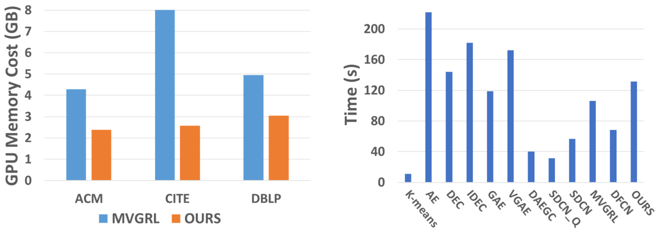
Figure 8: Sensitivity Analysis of hyper-parameter K. The results on DBLP, CITE, ACM (sub-figure a) and AMAP, PUBMED, CORAFULL (sub-figure b) datasets are illustrated.

4.5.2 *Sensitivity Analysis of Hyper-parameter K*

The sensitivity of hyper-parameter K in IDCRN is explored. Fig. 8 shows that the accuracy of IDCRN firstly increases to the high peek value and then stays at it with the slight perturbation as K increases. Besides, our proposed method is robust to the variation of K.

4.6 GPU Memory Costs and Time Costs

GPU memory and time costs are two important indicators for the algorithm evaluation. Compared to other contrastive-learning-base methods, IDCRN could save GPU memory costs since it eliminates the space-consuming negative sample generation. In order to certify this advantage of IDCRN, we conduct experiments of GPU memory costs and report the average results on the ACM, CITE, and DBLP datasets in Fig. 9 (a). From the results, we observe that IDCRN saves about 53.55 % GPU memory against MVGRL [15] on average. Furthermore, we also test the algorithm running time of the baselines and IDCRN in Fig. 9 (b) on ACM dataset. From these results, we observe that our method has comparable time costs compared to other baselines. The running time of our proposed method could be optimized in the future.



(a) GPU memory costs

(b) Time costs

Figure 9: GPU memory (sub-figure a) and time cost (sub-figure b) comparison between our method and the state-of-the-art methods.

4.7 Visualization Experiments

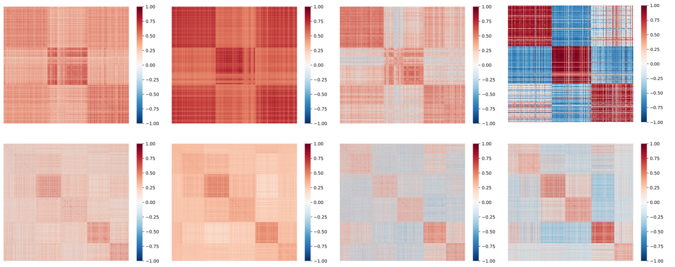
To intuitively show the superiority of IDCRN, two visualization experiments are conducted in this section.

4.7.1 *Visualization of Node Similarity Matrices*

We plot the heat maps of sample similarity matrices in the learned space to intuitively show the representation collapse problem in deep clustering methods and the effectiveness of our solution to this issue. The red, blue, and white colors indicate positive correlation, negative correlation, and decorrelation, respectively. Here, we sort all samples by categories to make those from the same cluster beside each other. As illustrated in Fig. 10, we observe that GAE [14] and MVGRL [15] 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.

4.7.2 *t-SNE Visualization of the Learned Embeddings*

In addition, we utilize t-SNE algorithm [52] to visualize the node embeddings $\mathbf{Z}$ learned by AE [19], DEC [30], GAE [14], ARGA [3], DFCN [11] and our proposed IDCRN. The t-SNE algorithm is a non-linear dimensionality reduction algorithm based on t-distribution. From the results as illustrated in Fig. 4, we observe that IDCRN learns a clearer structure of distribution in the latent space, thus better revealing the intrinsic clustering structure among the graph data. Besides, we further show the process of training our proposed method on DBLP, CITE, ACM, and AMAP datasets by performing t-SNE algorithm [52] over the learned node embeddings $\mathbf{Z}$ per 80 training epochs. From these results in Fig. 11, we observe that the distribute structure of learned node embeddings becomes clearer when the number of training epoch increases.



(a) GAE (b) MVGRL (c)SDCN (d) IDCRN

Figure 10: Visualization of sample similarity matrices in the latent space learned by our proposed method (IDCRN), SDCN [9], MVGRL [15], and GAE [14] on two benchmarks. The first row and second row correspond to the results on ACM and CITE datasets, respectively.

5 Conclusion

In this paper, to solve the representation collapse problem, we propose a novel deep graph clustering method termed Improved Dual Correlation Reduction Network (IDCRN) by improving the discriminative capability of node embeddings in the sample and feature aspects. Specifically, to the feature aspect, we reduce the redundancy between different dimensions of the learned features by approximating the cross-view feature correlation matrix to an identity matrix, thus improving the discriminative capability of the learned space explicitly. Simultaneously, in the sample aspect, we force the cross-view sample correlation matrix to approximate the designed clustering-refined adjacency matrix. With this setting, we guide the learned latent representation to recover the affinity matrix even across views, thus improving the feature discriminative capability implicitly. Extensive experimental results on six benchmarks demonstrate the effectiveness and efficiency of IDCRN. In the future, it is worth trying to apply IDCRN to more challenging applications, such as incomplete deep graph clustering.

Acknowledgments

This work was supported by the National Key R&D Program of China (project no. 2020AAA0107100) and the National Natural Science Foundation of China (project no. 62006237, 61922088, 61906020 and 61773392).

References

[1] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” *arXiv preprint arXiv:1609.02907*, 2016.

[2] C. Wang, S. Pan, R. Hu, G. Long, J. Jiang, and C. Zhang, “Attributed graph clustering: A deep attentional embedding approach,” *arXiv preprint arXiv:1906.06532*, 2019.

[3] S. Pan, R. Hu, S.-f. Fung, G. Long, J. Jiang, and C. Zhang, “Learning graph embedding with adversarial training methods,” *IEEE transactions on cybernetics*, vol. 50, no. 6, pp. 2475–2487, 2019.

[4] Z. Tao, H. Liu, J. Li, Z. Wang, and Y. Fu, “Adversarial graph embedding for ensemble clustering,” in *International Joint Conferences on Artificial Intelligence Organization*, 2019.

[5] B. Hui, P. Zhu, and Q. Hu, “Collaborative graph convolutional networks: Unsupervised learning meets semi-supervised learning,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 4215–4222.

[6] J. Park, M. Lee, H. J. Chang, K. Lee, and J. Y. Choi, “Symmetric graph convolutional autoencoder for unsupervised graph representation learning,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6519–6528.

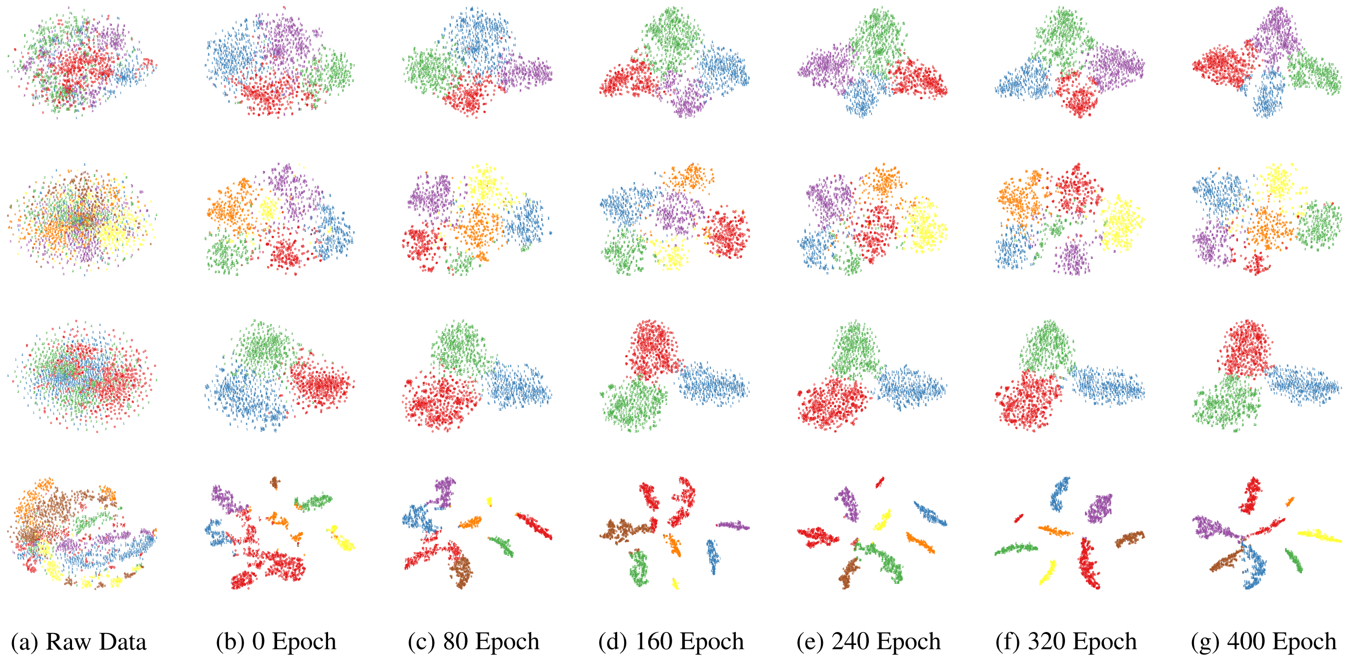


Figure 11: t-SNE [52] visualization of the raw data and the training process of our proposed method, including 0 (initialization), 80, 160, 240, 320, 400 epochs. In this figure, the first to the fourth row are the results on the DBLP, CITE, ACM, and AMAP datasets, respectively.

[7] G. Cui, J. Zhou, C. Yang, and Z. Liu, "Adaptive graph encoder for attributed graph embedding," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 976–985.

[8] S. Fan, X. Wang, C. Shi, E. Lu, K. Lin, and B. Wang, "One2multi graph autoencoder for multi-view graph clustering," in *Proceedings of The Web Conference 2020*, 2020, pp. 3070–3076.

[9] D. Bo, X. Wang, C. Shi, M. Zhu, E. Lu, and P. Cui, "Structural deep clustering network," in *Proceedings of The Web Conference 2020*, 2020, pp. 1400–1410.

[10] Z. Peng, H. Liu, Y. Jia, and J. Hou, "Attention-driven graph clustering network," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 935–943.

[11] W. Tu, S. Zhou, X. Liu, X. Guo, Z. Cai, E. Zhu, and J. Cheng, "Deep fusion clustering network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 11, 2021, pp. 9978–9987.

[12] Z. Peng, H. Liu, Y. Jia, and J. Hou, "Deep attention-guided graph clustering with dual self-supervision," *arXiv preprint arXiv:2111.05548*, 2021.

[13] N. Mrabah, M. Bouguessa, M. F. Touati, and R. Ksantini, "Rethinking graph auto-encoder models for attributed graph clustering," *arXiv preprint arXiv:2107.08562*, 2021.

[14] T. N. Kipf and M. Welling, "Variational graph auto-encoders," *arXiv preprint arXiv:1611.07308*, 2016.

[15] K. Hassani and A. H. Khasahmadi, "Contrastive multi-view representation learning on graphs," in *International Conference on Machine Learning*. PMLR, 2020, pp. 4116–4126.

[16] Y. Tian, X. Chen, and S. Ganguli, "Understanding self-supervised learning dynamics without contrastive pairs," in *International Conference on Machine Learning*. PMLR, 2021, pp. 10 268–10 278.

[17] H. Yang, K. Ma, and J. Cheng, "Rethinking graph regularization for graph neural networks," *arXiv preprint arXiv:2009.02027*, 2020.

[18] Y. Liu, W. Tu, S. Zhou, X. Liu, L. Song, X. Yang, and E. Zhu, "Deep graph clustering via dual correlation reduction," in *AAAI Conference on Artificial Intelligence*, 2022.

[19] B. Yang, X. Fu, N. D. Sidiropoulos, and M. Hong, "Towards k-means-friendly spaces: Simultaneous deep learning and clustering," in *international conference on machine learning*. PMLR, 2017, pp. 3861–3870.

[20] E. Pan and Z. Kang, "Multi-view contrastive graph clustering," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[21] X. Liu, F. Zhang, Z. Hou, L. Mian, Z. Wang, J. Zhang, and J. Tang, "Self-supervised learning: Generative or contrastive," *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[22] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9729–9738.

[23] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.

[24] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar *et al.*, "Bootstrap your own latent: A new approach to self-supervised learning," *arXiv preprint arXiv:2006.07733*, 2020.

[25] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," *arXiv preprint arXiv:2006.09882*, 2020.

[26] X. Chen and K. He, "Exploring simple siamese representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15 750–15 758.

[27] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, "Barlow twins: Self-supervised learning via redundancy reduction," *arXiv preprint arXiv:2103.03230*, 2021.

[28] A. Bardes, J. Ponce, and Y. Lecun, "Vicreg: Variance-invariance-covariance regularization for self-supervised learning," in *ICLR 2022-10th International Conference on Learning Representations*, 2022.

[29] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the royal statistical society. series c (applied statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[30] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *International conference on machine learning*. PMLR, 2016, pp. 478–487.

[31] X. Guo, L. Gao, X. Liu, and J. Yin, "Improved deep embedded clustering with local structure preservation." in *Ijcai*, 2017, pp. 1753–1759.

[32] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, "Graph contrastive learning with augmentations," *Advances in Neural Information Processing Systems*, vol. 33, pp. 5812–5823, 2020.

[33] Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, and L. Wang, "Graph contrastive learning with adaptive augmentation," in *Proceedings of the Web Conference 2021*, 2021, pp. 2069–2080.

[34] R. A. Horn, "The hadamard product," in *Proc. Symp. Appl. Math*, vol. 40, 1990, pp. 87–169.

[35] W. Tu, S. Zhou, Y. Liu, and X. Liu, "Siamese attribute-missing graph auto-encoder," *arXiv preprint arXiv:2112.04842*, 2021.

[36] O. Kramer, "K-nearest neighbors," in *Dimensionality reduction with unsupervised nearest neighbors*. Springer, 2013, pp. 13–23.

[37] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[38] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web." Stanford InfoLab, Tech. Rep., 1999.

[39] M. McPherson, L. Smith-Lovin, and J. M. Cook, "Birds of a feather: Homophily in social networks," *Annual review of sociology*, vol. 27, no. 1, pp. 415–444, 2001.

[40] B. Fuglede and F. Topsoe, "Jensen-shannon divergence and hilbert space embedding," in *International Symposium onInformation Theory, 2004. ISIT 2004. Proceedings.* IEEE, 2004, p. 31.

[41] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[42] J. McAuley, C. Targett, Q. Shi, and A. Van Den Hengel, "Image-based recommendations on styles and substitutes," in *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, 2015, pp. 43–52.

[43] G. Namata, B. London, L. Getoor, B. Huang, and U. EDU, "Query-driven active surveying for collective classification," 2012.

[44] A. Bojchevski and S. Günnemann, "Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking," in *International Conference on Learning Representations*, 2018, pp. 1–13.

[45] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[46] S. Liu, S. Wang, P. Zhang, X. Liu, K. Xu, C. Zhang, and F. Gao, "Efficient one-pass multi-view subspace clustering with consensus anchors," in *AAAI Conference on Artificial Intelligence*, 2022.

[47] S. Zhou, X. Liu, M. Li, E. Zhu, L. Liu, C. Zhang, and J. Yin, "Multiple kernel clustering with neighbor-kernel subspace segmentation," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 4, pp. 1351–1362, 2019.

[48] S. Zhou, E. Zhu, X. Liu, T. Zheng, Q. Liu, J. Xia, and J. Yin, "Subspace segmentation-based robust multiple kernel clustering," *Information Fusion*, vol. 53, pp. 145–154, 2020.

[49] T. Zhang, X. Liu, L. Gong, S. Wang, X. Niu, and L. Shen, "Late fusion multiple kernel clustering with local kernel alignment maximization," *IEEE Transactions on Multimedia*, pp. 1–1, 2021.

[50] S. Wang, X. Liu, E. Zhu, C. Tang, J. Liu, J. Hu, J. Xia, and J. Yin, "Multi-view clustering via late fusion alignment maximization." in *IJCAI*, 2019, pp. 3778–3784.

[51] M. D. Plummer and L. Lovász, *Matching theory*. Elsevier, 1986.

[52] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

[53] G. J. McLachlan and T. Krishnan, *The EM algorithm and extensions*. John Wiley & Sons, 2007, vol. 382.

Yue Liu graduated from Northeastern University at Qinhuangdao, Hebei, China. He was recommended for admission to the National University of Defense Technology (NUDT) with excellent grades and technological innovation capability. He is working hard and pursuing his master degree in College of Computer, NUDT, China. His current research interests include graph neural networks, deep clustering and self-supervised learning.

Sihang Zhou received his PhD degree from School of Computer, National University of Defense Technology (NUDT), China. He is now lecturer at College of Intelligence Science and Technology, NUDT. His current research interests include machine learning and medical image analysis. Dr. Zhou has published 40+ peer-reviewed papers, including IEEE T-IP, IEEE T-NNLS, IEEE T-MI, Information Fusion, Medical Image Analysis, AAAI, MICCAI, etc.

Xinwang Liu received his PhD degree from National University of Defense Technology (NUDT), China. He is now Professor of School of Computer, NUDT. His current research interests include kernel learning and unsupervised feature learning. Dr. Liu has published 60+ peer-reviewed papers, including those in highly regarded journals and conferences such as IEEE T-PAMI, IEEE T-KDE, IEEE T-IP, IEEE T-NNLS, IEEE T-MM, IEEE T-IFS, ICML, NeurIPS, ICCV, CVPR, AAAI, IJCAI, etc. He serves as the associated editor of Information Fusion Journal. More information can be found at https://xinwangliu.github.io/.

Wenxuan Tu is pursuing his Ph.D. degree in College of Computer, National University of Defense Technology (NUDT), China. His research interests include unsupervised graph learning, deep graph clustering, and image semantic segmentation. He has published several papers in highly regarded journals and conferences such as AAAI, ICML, MM, IEEE T-IP, Information Sciences, etc.

Xihong Yang is recommended for admission to the National University of Defense Technology (NUDT) as a master's student with excellent grades and competition awards. He is working hard to pursue his master degree. His current research interests include semi-supervised learning, self-supervised learning and graph neural networks.

Xin Xu received the B.S. degree in electrical engineering from the Department of Automatic Control, National University of Defense Technology (NUDT), Changsha, P. R. China, in 1996 and the Ph.D. degree in control science and engineering from the College of Mechatronics and Automation (CMA), NUDT. He has been a visiting scientist for cooperation research in the Hong Kong Polytechnic University, University of Alberta, and the University of Strathclyde, respectively. Currently, he is a full professor with the College of Intelligence Science and Technology and Director of the Department of Artificial Intelligence, National University of Defense Technology, Changsha, P.R. China. Prof. Xu's main research fields include machine learning and autonomous control of robots and intelligent unmanned systems. He received the Distinguished Young Scholars' Funds of National Natural Science Foundation of China. He is one of the recipients of the second-class National Natural Science Award of China and 2 first-class Natural Science Awards of Hunan Province, China. He has published 2 monographs and more than 200 papers. He is a senior member of IEEE and an associate editor of IEEE Transactions on System, Man and Cybernetics: Systems, Information Sciences, International Journal of Robotics and Automation, associate Editor-in-Chief of CAAI transactions on Intelligence Technology, and an Editorial Board Member of the Journal of Control Theory and Applications.

Fuchun Sun is a full professor of department of computer science and technology, Tsinghua University, IEEE/CAAI/CAA Fellow. He serves as Vice Chairman of Chinese Association for Artificial Intelligence and Executive Director of Chinese Association for Automation. His research interests include robotic perception and skill learning, cross-modal learning and robot dexterous operations. Dr. Sun is the recipient of the excellent Doctoral Dissertation Prize of China in 2000 by MOE of China and the Choon-Gang Academic Award by Korea in 2003, and was recognized as a Distinguished Young Scholar in 2006 by the Natural Science Foundation of China. He served as the EIC of the Journal of Cognitive Computation and Systems, and associated editors of IEEE Trans. on Neural Networks and Learning Systems during 2006-2010, IEEE Trans. On Fuzzy Systems since 2011, IEEE Trans. on Cognitive and Development Systems since 2018 and IEEE Trans. on Systems, Man and Cybernetics: Systems since 2015.

Deep Graph Clustering via Dual Correlation Reduction

Yue Liu,[1*] Wenxuan Tu,[1*] Sihang Zhou,[2] Xinwang Liu,[1†]
Linxuan Song,[1] Xihong Yang,[1] En Zhu[1]

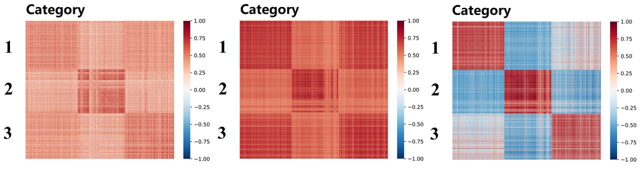
[1]College of Computer, National University of Defense Technology, Changsha, China
[2]College of Intelligence Science and Technology, National University of Defense Technology, Changsha, China
{yueliu, twx, xinwangliu, yangxihong, enzhu}@nudt.edu.cn, sihangjoe@gmail.com, slxnatavidad@163.com

Abstract

Deep graph clustering, which aims to reveal the underlying graph structure and divide the nodes into different groups, has attracted intensive attention in recent years. However, we observe that, in the process of node encoding, existing methods suffer from representation collapse which tends to map all data into the same representation. Consequently, the discriminative capability of the node representation is limited, leading to unsatisfied clustering performance. To address this issue, we propose a novel self-supervised deep graph clustering method termed **D**ual **C**orrelation **R**eduction **N**etwork (**DCRN**) by reducing information correlation in a dual manner. Specifically, in our method, we first design a siamese network to encode samples. Then by forcing the cross-view sample correlation matrix and cross-view feature correlation matrix to approximate two identity matrices, respectively, we reduce the information correlation in the dual-level, thus improving the discriminative capability of the resulting features. Moreover, in order to alleviate representation collapse caused by over-smoothing in GCN, we introduce a propagation regularization term to enable the network to gain long-distance information with the shallow network structure. Extensive experimental results on six benchmark datasets demonstrate the effectiveness of the proposed DCRN against the existing state-of-the-art methods. *The code of DCRN is available at DCRN and a collection of deep graph clustering is shared at Awesome Deep Graph Clustering on Github.*

Introduction

Deep graph clustering is a fundamental yet challenging task whose target is to train a neural network for learning representations to divide nodes into different groups without human annotations. Thanks to the powerful graph information exploitation capability, graph convolutional networks (GCN) (Kipf and Welling 2016a) have recently achieved promising performance in many graph clustering applications like social networks and recommendation systems. Consequently, it has attracted considerable attention in this field and many algorithms are proposed (Wang et al. 2019; Pan et al. 2019; Tao et al. 2019; Park et al. 2019; Bo et al. 2020; Tu et al. 2020).



(a) GAE (b) MVGRL (c) OURS

Figure 1: The heat maps of node similarity matrices in the latent space of GAE (Kipf and Welling 2016b), MVGRL (Hassani and Khasahmadi 2020), and our proposed method on the ACM dataset.

Though good performance has been achieved, we found that the existing GCN-based clustering algorithms usually suffer from the representation collapse problem and tend to map nodes from different categories into the similar representation in the process of sample encoding. As a result, the node representation is indiscriminative and the clustering performance is limited. We illustrate this phenomenon on ACM dataset in Fig. 1. In this figure, we first extract the node embedding learned from three representative algorithms, i.e., the Graph Auto-Encoder (GAE) (Kipf and Welling 2016b), Multi-View Graph Representation Learning (MVGRL) (Hassani and Khasahmadi 2020), and our proposed algorithm (OURS), and then construct the element-wise similarity matrices by calculating the cosine similarity, respectively. Finally, we visualize the similarity matrices of the three compared algorithms in Fig. 1. Among the compared algorithms, GAE is a classic graph convolutional network, MVGRL is a contrastive strategy enhanced algorithm, which can to some extent alleviate the representation collapse problem by introducing a positive and negative sample pair recognition mechanism. From sub-figure (a) and (b), we observe that, in the latent space learned by both the classic algorithm and the contrastive learning enhanced algorithm, the intrinsic three dimensional cluster space is not well revealed. It indicates that representation collapse is still an open problem which is restricting the performance of GCN-based clustering algorithms.

To solve this problem, we propose a novel self-supervised deep graph clustering method termed Dual Correlation Reduction Network (DCRN) to avoid representation collapse by reducing the information correlation in a dual manner. To be specific, in our network, a dual information correla-

[*]First author with equal contribution

[†]Corresponding author

tion reduction mechanism is introduced to force the cross-view sample correlation matrix and cross-view feature correlation matrix to approximate two identity matrices, respectively. In this setting, by forcing the cross-view sample-level correlation matrix to approximate an identical matrix, we guide the same noise-disturbed samples to have the identical representation while different samples to have the different representation. In this way, the sample representations would be more discriminative and in the meantime more robust against noisy information. Similarly, by letting the cross-view feature-level correlation matrix to approximate an identical matrix, the discriminative capability of latent feature is enhanced since different dimensions of the latent feature are decorrelated. This could be clearly seen in Fig. 1 (c) since the similarity matrix generated by our proposed method can obviously exploit the hidden cluster structure among data better than the compared algorithms. As a self-supervised method, since our algorithm gets rid of the complex and space-consuming negative sample construction operations, it is more space-saving than the other contrastive learning-based algorithms. For example, in the process of model training with all samples on DBLP, CITE and ACM datasets, MVGRL spends 5753M GPU memory on average while our proposed method only spends 2672M on average. Moreover, motivated by propagation regularization (Yang, Ma, and Cheng 2020), in order to alleviate representation collapse caused by over-smoothing in GCN (Kipf and Welling 2016a), we improve the long-distance information capture capability of our model with shallow network structure by introducing a propagation regularization term. This further improves the clustering performance of our proposed algorithm. The key contributions of this paper are listed as follows.

- We propose a siamese network-based algorithm to solve the problem of representation collapse in the field of deep graph clustering.
- A dual correlation reduction strategy is proposed to improve the discriminative capability of the sample representation. Thanks to this strategy, our method is free from the complicated negative sample generation operation and thus is more space-saving and more flexible against training batch size.
- Extensive experimental results on six benchmark datasets demonstrate the superiority of the proposed method against the existing state-of-the-art deep graph clustering competitors.

Related Work

Attributed Graph Clustering

Graph Neural Networks (GNNs), which learn the representation from both node attributes and graph structures, have emerged as a powerful approach for attributed graph clustering. Specifically, GAE/VGAE (Kipf and Welling 2016b) embeds the node attributes with structure information via a graph encoder and then reconstructs the graph structure by an inner product decoder. Inspired by their success, recent researches, DAEGC (Wang et al. 2019), GALA (Park et al. 2019), ARGA (Pan et al. 2019) and AGAE (Tao et al. 2019) further improve the early works with graph attention network, Laplacian sharpening, and generative adversarial learning. Although achieving promising clustering performance, the over-smoothing problem has not been effectively tackled in these methods, which affects the clustering performance. More recently, SDCN (Bo et al. 2020) and DFCN (Tu et al. 2020) are proposed to jointly learn an Auto-Encoder (AE) (Yang et al. 2017) and a Graph Auto-Encoder (GAE) (Kipf and Welling 2016b) in a united framework to alleviate the over-smoothing problem via an information transport operation and a structure-attribute fusion module, respectively. Although both methods have proved that introducing the attribute features into the latent structure space can effectively address the over-smoothing issue, SDCN and DFCN suffer from another non-negligible limitation, i.e., information correlation, resulting in less discriminative representations and sub-optimal clustering performance. In contrast, our method improves the existing advanced deep graph clustering algorithm by introducing a dual information correlation reduction mechanism from the perspective of sample and feature levels to alleviate representation collapse.

Representation Collapse

Representation collapse, which maps all data into a same representation, is a common issue in current self-supervised representation learning methods. Some contrastive learning methods are proposed to solve this problem. MoCo (He et al. 2020) utilizes a momentum encoder to maintain the consistent representation of negative pairs drawn from a memory bank. SimCLR (Chen et al. 2020) defines the “positive” and “negative” sample pairs, and pulls closer the “positive” samples existing in the current batch while pushing the “negative” samples away. By replacing the empty cluster with a perturbated non-empty cluster, DeepCluster (Caron et al. 2018) is able to alleviate the collapsed representation. In addition, BYOL (Grill et al. 2020) and SimSiam (Chen and He 2021) have demonstrated that the momentum encoder and the stop-gradient mechanism are crucial to avoid representation collapse without demanding negative samples for producing prediction targets. More recently, a simple yet effective algorithm, Barlow Twins (Zbontar et al. 2021) is proposed to alleviate the collapsed representation by reducing the redundant information between the representation of distorted samples. Inspired by its advantages, we naturally extend the idea of Barlow Twins into deep graph clustering and further design a dual correlation reduction mechanism to address representation collapse in deep clustering network. Compared to other contrastive learning methods, our proposed method learns the discriminative embedding to avoid collapse without negative sample generation, large batches or asymmetric mechanisms.

Dual Correlation Reduction Network

We introduce a novel self-supervised deep graph clustering method termed Dual Correlation Reduction Network (DCRN), which aims to avoid representation collapse by reducing information correlation in a dual manner. As illustrated in Fig. 2, DCRN mainly consists of two components,

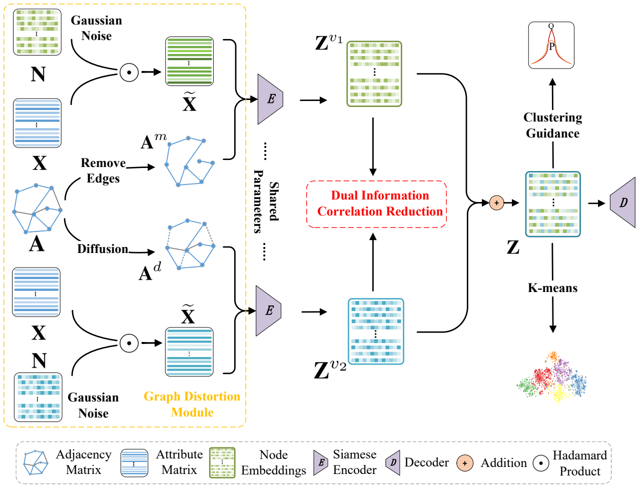


Figure 2: Illustration of the Dual Correlation Reduction Network (DCRN). In the proposed algorithm, the graph distortion module first generates two distorted graphs by introducing attribute and graph disturbances. Then, by forcing the same sample within two distorted graphs to have identical representations in both feature level and sample level, while different samples have different representations also in dual levels, the network is guided to be more discriminative with less memory consumption.

i.e., a graph distortion module and a dual information correlation reduction (DICR) module. Note that the extraction backbone network of DCRN is similar to that of DFCN (Tu et al. 2020). In the following sections, We will introduce the graph distortion module, DICR module, and network objectives in detail.

Notations and Problem Definition

Given an undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with C categories of nodes, $\mathcal{V} = \{v_1, v_2, \dots, v_N\}$ and $\mathcal{E}$ are the node set and the edge set, respectively. The graph is characterized by its attribute matrix $\mathbf{X} \in \mathbb{R}^{N \times D}$ and original adjacency matrix $\mathbf{A} = (a_{ij})_{N \times N}$, where $a_{ij} = 1$ if $(v_i, v_j) \in \mathcal{E}$, otherwise $a_{ij} = 0$. The corresponding degree matrix is $\mathbf{D} = diag(d_1, d_2, \dots, d_N) \in \mathbb{R}^{N \times N}$ and $d_i = \sum_{(v_i, v_j) \in \mathcal{E}} a_{ij}$. With $\mathbf{D}$, the original adjacency matrix $\mathbf{A}$ can be normalized as $\widetilde{\mathbf{A}} \in \mathbb{R}^{N \times N}$ through calculating $\mathbf{D}^{-1}(\mathbf{A} + \mathbf{I})$, where $\mathbf{I} \in \mathbb{R}^{N \times N}$ is an identity matrix. In this paper, we aim to train a siamese graph encoder that embeds all nodes into the low-dimension latent space in an unsupervised manner. The resultant latent embedding can then be directly utilized to perform node clustering by K-means (Hartigan and Wong 1979). The notations are summarized in Table 1.

Graph Distortion Module

Recent efforts in self-supervised graph representation learning have demonstrated that graph distortion could enable the network to learn rich representations from different contexts for nodes (Hassani and Khasahmadi 2020; You et al. 2020). Inspired by their success, as illustrated in Fig. 2, we consider two types of distortion on graphs, i.e., feature corruption and edge perturbation.

Notations	Meaning
$\mathbf{X} \in \mathbb{R}^{N \times D}$	Attribute matrix
$\mathbf{A} \in \mathbb{R}^{N \times N}$	Original adjacency matrix
$\mathbf{D} \in \mathbb{R}^{N \times N}$	Degree matrix
$\widetilde{\mathbf{A}} \in \mathbb{R}^{N \times N}$	Normalized adjacency matrix
$\mathbf{A}^m \in \mathbb{R}^{N \times N}$	Edge-masked adjacency matrix
$\mathbf{A}^d \in \mathbb{R}^{N \times N}$	Graph diffusion matrix
$\widehat{\mathbf{X}} \in \mathbb{R}^{N \times D}$	Rebuilt attribute matrix
$\widehat{\mathbf{A}} \in \mathbb{R}^{N \times N}$	Rebuilt adjacency matrix
$\mathbf{Z}^{v_k} \in \mathbb{R}^{N \times d}$	Node embedding in k-th view
$\mathbf{Z} \in \mathbb{R}^{N \times d}$	Clustering-oriented latent embedding
$\widetilde{\mathbf{Z}}^{v_k} \in \mathbb{R}^{d \times K}$	Cluster-level embedding in k-th view
$\mathbf{S}^{\mathcal{N}} \in \mathbb{R}^{N \times N}$	Cross-view sample correlation matrix
$\mathbf{S}^{\mathcal{F}} \in \mathbb{R}^{d \times d}$	Cross-view feature correlation matrix
$\mathbf{Q} \in \mathbb{R}^{N \times C}$	Soft assignment distribution
$\mathbf{P} \in \mathbb{R}^{N \times C}$	Target distribution

Table 1: Notation summary

Feature Corruption. For the attribute-level distortion, we first sample a random noise matrix $\mathbf{N} \in \mathbb{R}^{N \times D}$ from a Gaussian distribution $\mathcal{N}(1, 0.1)$. Then the resulting corrupted attribute matrix $\widetilde{\mathbf{X}} \in \mathbb{R}^{N \times D}$ can be formulated:

$$\widetilde{\mathbf{X}} = \mathbf{X} \odot \mathbf{N}, \tag{1}$$

where $\odot$ denotes the Hadamard product (Horn 1990).

Edge Perturbation. In addition to corrupting node features, for structure-level distortion, we introduce two strategies for edge perturbation. One is similarity-based edge removing. Thus, we first calculate the sample pair-wise cosine similarity in latent space, and then generate a masked matrix $\mathbf{M} \in \mathbb{R}^{N \times N}$ according to the similarity matrix, where the lowest 10% linkage relation would be manually removed. Finally, the edge-masked adjacency matrix $\mathbf{A}^m \in \mathbb{R}^{N \times N}$ would be normalized and be computed as:

$$\mathbf{A}^m = \mathbf{D}^{-\frac{1}{2}}((\mathbf{A} \odot \mathbf{M}) + \mathbf{I})\mathbf{D}^{-\frac{1}{2}}. \tag{2}$$

The other is the graph diffusion, where we follow MVGRL (Hassani and Khasahmadi 2020) to transform the normalized adjacency matrix to a graph diffusion matrix by Personalized PageRank (PPR) (Page et al. 1999):

$$\mathbf{A}^d = \alpha(\mathbf{I} - (1 - \alpha)(\mathbf{D}^{-\frac{1}{2}}(\mathbf{A} + \mathbf{I})\mathbf{D}^{-\frac{1}{2}}))^{-1}, \tag{3}$$

where α is the teleport probability that is set to 0.2. Finally, we denote $\mathcal{G}^1 = (\widetilde{\mathbf{X}}, \mathbf{A}^m)$ and $\mathcal{G}^2 = (\widetilde{\mathbf{X}}, \mathbf{A}^d)$ as two views of the graph, respectively.

Dual Information Correlation Reduction

In this section, we introduce a dual information correlation reduction (DICR) mechanism to filter the redundant information of the latent embedding in a dual manner, i.e., sample-level correlation reduction (SCR) and feature-level correlation reduction (FCR), which aims to constrain our network to learn more discriminative latent features, thus alleviating representation collapse. SCR and FCR are both illustrated in Fig. 3 in detail.

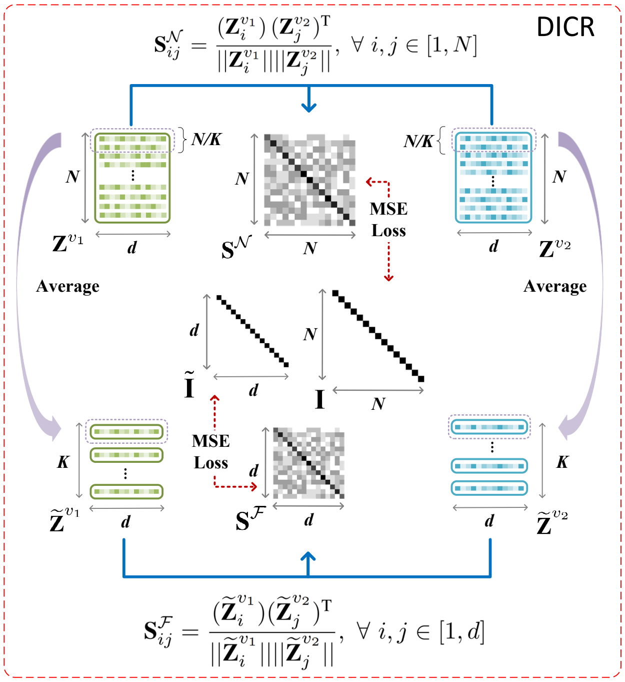

DICR
$S_{ij}^{\mathcal{N}} = \frac{(\mathbf{Z}_i^{v_1})(\mathbf{Z}_j^{v_2})^{\mathrm{T}}}{||\mathbf{Z}_i^{v_1}||||\mathbf{Z}_j^{v_2}||}, \forall\, i,j \in [1,N]$
N/K
N
d
$\mathbf{Z}^{v_1}$
$\mathbf{S}^{\mathcal{N}}$
MSE Loss
$\mathbf{Z}^{v_2}$
Average
Average
$\widetilde{\mathbf{I}}$
$\mathbf{I}$
K
$\widetilde{\mathbf{Z}}^{v_1}$
$\mathbf{S}^{\mathcal{F}}$
$\widetilde{\mathbf{Z}}^{v_2}$
$S_{ij}^{\mathcal{F}} = \frac{(\widetilde{\mathbf{Z}}_i^{v_1})(\widetilde{\mathbf{Z}}_j^{v_2})^{\mathrm{T}}}{||\widetilde{\mathbf{Z}}_i^{v_1}||||\widetilde{\mathbf{Z}}_j^{v_2}||}, \forall\, i,j \in [1,d]$

Figure 3: Illustration of Dual Information Correlation Reduction (DICR) mechanism.

Sample-level Correlation Reduction. The learning process of SCR includes two steps. For given two-view node embeddings $\mathbf{Z}^{v_1}$ and $\mathbf{Z}^{v_2}$ learned by a siamese graph encoder, we firstly calculate the elements in cross-view sample correlation matrix $\mathbf{S}^{\mathcal{N}} \in \mathbb{R}^{N\times N}$ by:

$$\mathbf{S}_{ij}^{\mathcal{N}} = \frac{(\mathbf{Z}_i^{v_1})\,(\mathbf{Z}_j^{v_2})^{\mathrm{T}}}{||\mathbf{Z}_i^{v_1}||||\mathbf{Z}_j^{v_2}||},\ \forall\ i,j \in [1,N], \tag{4}$$

where $\mathbf{S}_{ij}^{\mathcal{N}} \in [-1,1]$ denotes the cosine similarity between i-th node embedding in the first view and j-th node embedding in the second view. After that, we make the cross-view sample correlation matrix $\mathbf{S}^{\mathcal{N}}$ to be equal to an identity matrix $\mathbf{I} \in \mathbb{R}^{N\times N}$, formulated as:

$$\begin{aligned}\mathcal{L}_N &= \frac{1}{N^2}\sum(\mathbf{S}^{\mathcal{N}} - \mathbf{I})^2\\ &= \frac{1}{N}\sum_{i=1}^{N}\left(\mathbf{S}_{ii}^{\mathcal{N}} - 1\right)^2 + \frac{1}{N^2-N}\sum_{i=1}^{N}\sum_{j\neq i}\left(\mathbf{S}_{ij}^{\mathcal{N}}\right)^2,\end{aligned} \tag{5}$$

where the first term encourages the diagonal elements in $\mathbf{S}^{\mathcal{N}}$ equal to 1, which indicates that the embedding of each node in two different views are enforced to agree with each other. The second term makes the off-diagonal elements in $\mathbf{S}^{\mathcal{N}}$ equal to 0 to minimize the agreement between embeddings of different nodes across two views. This decorrelation operation could help our network reduce the redundant information among nodes in the latent space so that the learned embedding could be more discriminative.

Feature-level Correlation Reduction. Apart from building nontrivial embeddings by reducing the sample correlation across two views, we further consider to refine the information correlation from the aspect of feature dimension. Specifically, Fig. 3 illustrates our feature-level correlation reduction design, which is implemented in three steps. First, we project two-view node embeddings $\mathbf{Z}^{v_1}$ and $\mathbf{Z}^{v_2}$ into cluster-level embeddings $\widetilde{\mathbf{Z}}^{v_1}$ and $\widetilde{\mathbf{Z}}^{v_2} \in \mathbb{R}^{d\times K}$ using a readout function $\mathcal{R}(\cdot) : \mathbb{R}^{d\times N} \to \mathbb{R}^{d\times K}$, formulated as:

$$\widetilde{\mathbf{Z}}^{v_k} = \mathcal{R}\left((\mathbf{Z}^{v_k})^{\mathrm{T}}\right). \tag{6}$$

Then we again calculate the cosine similarity between $\widetilde{\mathbf{Z}}^{v_1}$ and $\widetilde{\mathbf{Z}}^{v_2}$ along with the feature dimension, that's:

$$\mathbf{S}_{ij}^{\mathcal{F}} = \frac{(\widetilde{\mathbf{Z}}_i^{v_1})(\widetilde{\mathbf{Z}}_j^{v_2})^{\mathrm{T}}}{||\widetilde{\mathbf{Z}}_i^{v_1}||||\widetilde{\mathbf{Z}}_j^{v_2}||},\ \forall\ i,j \in [1,d], \tag{7}$$

where $\mathbf{S}_{ij}^{\mathcal{F}}$ denotes the feature similarity between i-th dimension feature in one view and j-th dimension in another view. Thereafter, similar to the objective functions Eq. (5), we make the cross-view feature correlation matrix $\mathbf{S}^{\mathcal{F}}$ to be equal to an identity matrix $\widetilde{\mathbf{I}} \in \mathbb{R}^{d\times d}$:

$$\begin{aligned}\mathcal{L}_F &= \frac{1}{d^2}\sum(\mathbf{S}^{\mathcal{F}} - \widetilde{\mathbf{I}})^2\\ &= \frac{1}{d^2}\sum_{i=1}^{d}\left(\mathbf{S}_{ii}^{\mathcal{F}} - 1\right)^2 + \frac{1}{d^2-d}\sum_{i=1}^{d}\sum_{j\neq i}\left(\mathbf{S}_{ij}^{\mathcal{F}}\right)^2,\end{aligned} \tag{8}$$

where d is the latent embedding dimension. Both terms in Eq. (8) mean that the representations of the same dimension feature in two augmented views are pulled closer while others are pushed away, respectively. Finally, we combine the decorrelated latent embeddings from two views with a linear combination operation, thus the resultant clustering-oriented latent embeddings $\mathbf{Z} \in \mathbb{R}^{N\times d}$ can then be used to performed clustering by K-means (Hartigan and Wong 1979):

$$\mathbf{Z} = \frac{1}{2}(\mathbf{Z}^{v_1} + \mathbf{Z}^{v_2}). \tag{9}$$

Technically, the proposed DICR mechanism considers the correlation reduction in both the perspective of the sample and feature level. In this way, the redundant features could be filtered while more discriminative features could be preserved in the latent space, thus the network can learn meaningful representations to avoid collapse for clustering performance improvement.

Propagated Regularization. Furthermore, in order to alleviate the over-smoothing phenomenon during the network training, we introduce a propagation regularization formulated as:

$$\mathcal{L}_R = JSD(\mathbf{Z},\ \widetilde{\mathbf{A}}\mathbf{Z}), \tag{10}$$

where $JSD(\cdot)$ refers to the Jensen-Shannon divergence (Fuglede and Topsoe 2004). With Eq. (10), the network is able to capture long-distance information with shallow network structure to alleviate over-smoothing when the propagated information goes deeper throughout the framework. In summary, the objective of DICR module can be computed by:

$$\mathcal{L}_{DICR} = \mathcal{L}_N + \mathcal{L}_F + \gamma\mathcal{L}_R, \tag{11}$$

where γ is a balanced hyper-parameter.

Algorithm 1: Dual Correlation Reduction Network

```
Input: Two-view graphs: $\mathcal{G}^1 = (\widetilde{\mathbf{X}}, \mathbf{A}^m)$, $\mathcal{G}^2 = (\widetilde{\mathbf{X}}, \mathbf{A}^d)$; Cluster number $C$; Iteration number $I$; Hyper-parameters $\gamma$ and $\lambda$.
Output: The clustering result $\mathbf{R}$.
Pre-train the baseline network to obtain $\mathbf{Z}$;
Initialize the cluster centers $u$ with K-means over $\mathbf{Z}$;
for $i = 1$ to $I$ do
    Utilize the baseline network to encode $\mathbf{Z}^{v_1}$ and $\mathbf{Z}^{v_2}$;
    Calculate $\widetilde{\mathbf{Z}}^{v_1}$ and $\widetilde{\mathbf{Z}}^{v_2}$ by Eq. (6);
    Calculate $\mathbf{S}^{\mathcal{N}}$ and $\mathbf{S}^{\mathcal{F}}$ by Eq. (4) and Eq. (7), respectively;
    Conduct the sample-level and the feature-level correlation reduction by Eq. (4) and Eq. (7), respectively;
    Fuse $\mathbf{Z}^{v_1}$ and $\mathbf{Z}^{v_2}$ to obtain $\mathbf{Z}$ by Eq. (9);
    Calculate $\mathcal{L}_{DICR}$, $\mathcal{L}_{REC}$, and $\mathcal{L}_{KL}$, respectively.
    Update the whole network by minimizing $\mathcal{L}$ in Eq. (12);
end for
Obtain $\mathbf{R}$ by performing K-means over $\mathbf{Z}$.
return $\mathbf{R}$
```

Objective Function

The overall optimization objective of the proposed method consists of three parts: the loss of proposed DICR, the reconstruction loss, and the clustering loss:

$$\mathcal{L} = \mathcal{L}_{DICR} + \mathcal{L}_{REC} + \lambda \mathcal{L}_{KL}, \tag{12}$$

where $\mathcal{L}_{REC}$ denotes the joint mean square error (MSE) reconstruction loss of node attributes and graph structure adopted in (Tu et al. 2020). $\mathcal{L}_{KL}$ denotes the Kullback–Leibler divergence (Kullback and Leibler 1951), i.e., a widely-used self-supervised clustering loss (Xie, Girshick, and Farhadi 2016; Guo et al. 2017; Wang et al. 2019; Bo et al. 2020; Tu et al. 2020), where we generate the soft assignment distribution $\mathbf{Q} \in \mathbb{R}^{N \times C}$ and the target distribution $\mathbf{P} \in \mathbb{R}^{N \times C}$ over the clustering-oriented node embeddings $\mathbf{Z}$, and then align both distributions to guide the network learning. The trade-off parameter λ is set to 10. Here, for the design of $\mathcal{L}_{REC}$ and $\mathcal{L}_{KL}$, more details are described in the origin paper of DFCN (Tu et al. 2020). The detailed learning procedure of DCRN is shown in Algorithm 1.

Expriments

Datasets

To evaluate the effectiveness of the proposed method, we conduct extensive experiments on six widely-used datasets, including DBLP, CITE, ACM(Bo et al. 2020), AMAP, PUBMED, and CORAFULL(Shchur et al. 2018). The brief information of these datasets is summarized in Table 2.

Experiment Setup

Training Procedure The proposed DCRN is implemented with a NVIDIA 3090 GPU on PyTorch platform. The training process of our model includes three steps. Following DFCN (Tu et al. 2020), we first pre-train the sub-networks independently with at least 30 epochs by minimizing the reconstruction loss $\mathcal{L}_{REC}$. Then both sub-networks are directly integrated into a united framework to obtain the initial clustering centers for another 100 epochs. Thereafter, we train the whole network under the guidance of Eq. (12) for 400 epochs until convergence. Finally, we perform clustering over $\mathbf{Z}$ by K-means (Hartigan and Wong 1979). To avoid randomness, we run each method for 10 times and report the averages with standard deviations.

Dataset	Samples	Dimension	Edges	Classes
DBLP	4057	334	3528	4
CITE	3327	3703	4552	6
ACM	3025	1870	13128	3
AMAP	7650	745	119081	8
PUBMED	19717	500	44325	3
CORAFULL	19793	8710	63421	70

Table 2: Dataset summary

Parameters Setting For ARGA/ARVGA (Pan et al. 2019), MVGRL (Hassani and Khasahmadi 2020), and DFCN (Tu et al. 2020), we reproduce their source code by following the setting of the original literature and present the average results. For other compared baselines, we directly report the corresponding values listed in DFCN (Tu et al. 2020). For our proposed method, we adopt the code and data of DFCN for data pre-processing and testing. Besides, we adopt DFCN (Tu et al. 2020) as our backbone network. The network is trained with the Adam optimizer(Kingma and Ba 2014) in all experiments. The learning rate is set to 1e-3 for AMAP, 1e-4 for DBLP, 5e-5 for ACM, 1e-5 for CITE, PUBMED, and CORAFULL, respectively. The hyper-parameters α is set to 0.1 for PUBMED and 0.2 for other datasets. Moreover, we set λ and γ to 10 and 1e3, respectively. K in Eq. 6 is set to the cluster number C.

Metrics The clustering performance is evaluated by four public metrics: Accuracy (ACC), Normalized Mutual Information (NMI), Average Rand Index (ARI) and macro F1-score (F1) (Liu et al. 2019a, 2018, 2019b; Zhou et al. 2019, 2020). The best map between cluster ID and class ID is constructed by the Kuhn-Munkres (Plummer and Lovász 1986).

Performance Comparison

To demonstrate the superiority of the proposed method, we adopt 13 baselines for performance comparisons. Specifically, K-means (Hartigan and Wong 1979) is one of the most classic traditional clustering methods. Three representative deep generative methods, i.e., AE (Yang et al. 2017), DEC (Xie, Girshick, and Farhadi 2016), and IDEC (Guo et al. 2017), train an auto-encoder and then perform a clustering algorithm over the learned latent embedding. GAE/VGAE (Kipf and Welling 2016b), DAEGC (Wang et al. 2019), and ARGA/ARVGA (Pan et al. 2019) are three typical GCN-based frameworks that learn the representation for clustering by considering both node attribute and structure information. Furthermore, we report the performance of three state-of-the-art deep clustering methods, i.e., SDCN/SDCN$_Q$ (Bo et al. 2020), DFCN (Tu et al. 2020), and MVGRL (Hassani and Khasahmadi 2020), which utilize two sub-networks to process augmented graphs independently.

Table 3 reports the clustering performance of all compared methods on six benchmarks. From these results, we can conclude that 1) DCRN consistently outperforms all

Dataset	Metric	K-Means	AE	DEC	IDEC	GAE	VGAE	DAEGC	ARGA	ARVGA	SDCN_Q	SDCN	MVGRL	DFCN	OURS
DBLP	ACC	38.65±0.65	51.43±0.35	58.16±0.56	60.31±0.62	61.21±1.22	58.59±0.06	62.05±0.48	64.83±0.59	54.41±0.42	65.74±1.34	68.05±1.81	42.73±1.02	76.00±0.80	79.66±0.25
	NMI	11.45±0.38	25.40±0.16	29.51±0.28	31.17±0.50	30.80±0.91	26.92±0.06	32.49±0.45	29.42±0.92	25.90±0.33	35.11±1.05	39.50±1.34	15.41±0.63	43.70±1.00	48.95±0.44
	ARI	6.97±0.39	12.21±0.43	23.92±0.39	25.37±0.60	22.02±1.40	17.92±0.07	21.03±0.52	27.99±0.91	19.81±0.42	34.00±1.76	39.15±2.01	8.22±0.50	47.00±1.50	53.60±0.46
	F1	31.92±0.27	52.53±0.36	59.38±0.51	61.33±0.56	61.41±2.23	58.69±0.07	61.75±0.67	64.97±0.66	55.37±0.40	65.78±1.22	67.71±1.51	40.52±1.51	75.70±0.80	79.28±0.26
CITE	ACC	39.32±3.17	57.08±0.13	55.89±0.20	60.49±1.42	61.35±0.80	60.97±0.36	64.54±1.39	61.07±0.49	59.31±1.38	61.67±1.05	65.96±0.31	68.66±0.36	69.50±0.20	70.86±0.18
	NMI	16.94±3.22	27.64±0.08	28.34±0.30	27.17±2.40	34.63±0.65	32.69±0.27	36.41±0.86	34.40±0.71	31.80±0.81	34.39±1.22	38.71±0.32	43.66±0.40	43.90±0.20	45.86±0.35
	ARI	13.43±3.02	29.31±0.14	28.12±0.36	25.70±2.65	33.55±1.18	33.13±0.53	37.78±1.24	34.32±0.70	31.28±1.22	35.50±1.49	40.17±0.43	44.27±0.73	45.50±0.30	47.64±0.30
	F1	36.08±3.53	53.80±0.11	52.62±0.17	61.62±1.39	57.36±0.82	57.70±0.49	62.20±1.32	58.23±0.31	56.05±1.13	57.82±0.98	63.62±0.24	63.71±0.39	64.30±0.20	65.83±0.21
ACM	ACC	67.31±0.71	81.83±0.08	84.33±0.76	85.12±0.52	84.52±1.44	84.13±0.22	86.94±2.83	86.29±0.36	83.89±0.54	86.95±0.08	90.45±0.18	86.73±0.76	90.90±0.20	91.93±0.20
	NMI	32.44±0.46	49.30±0.16	54.54±1.51	56.61±1.16	55.38±1.92	53.20±0.52	56.18±4.15	56.21±0.82	51.88±1.04	58.90±0.17	68.31±0.25	60.87±1.40	69.40±0.40	71.56±0.61
	ARI	30.60±0.69	54.64±0.16	60.64±1.87	62.16±1.50	59.46±3.10	57.72±0.67	59.35±3.89	63.37±0.86	57.77±1.17	65.25±0.19	73.91±0.40	65.07±1.76	74.90±0.40	77.56±0.52
	F1	67.57±0.74	82.01±0.08	84.51±0.74	85.11±0.48	84.65±1.33	84.17±0.23	87.07±2.79	86.31±0.35	83.87±0.55	86.84±0.09	90.42±0.19	86.85±0.72	90.80±0.20	91.94±0.20
AMAP	ACC	27.22±0.76	48.25±0.08	47.22±0.08	47.62±0.08	71.57±2.48	74.26±3.63	76.44±0.01	69.28±2.30	61.46±2.71	35.53±0.39	53.44±0.81	45.19±1.79	76.88±0.80	79.94±0.13
	NMI	13.23±1.33	38.76±0.30	37.35±0.05	37.83±0.08	62.13±2.79	66.01±3.40	65.57±0.03	58.36±2.76	53.25±1.91	27.90±0.40	44.85±0.83	36.89±1.31	69.21±1.00	73.70±0.24
	ARI	5.50±0.44	20.80±0.47	18.59±0.04	19.24±0.07	48.82±4.57	56.24±4.66	59.39±0.02	44.18±4.41	38.44±4.69	15.27±0.37	31.21±1.23	18.79±0.47	58.98±0.84	63.69±0.20
	F1	23.96±0.51	47.87±0.20	46.71±0.12	47.20±0.11	68.08±1.76	70.38±2.98	69.97±0.02	64.30±1.95	58.50±1.70	34.25±0.44	50.66±1.49	39.65±2.39	71.58±0.31	73.82±0.12
PUBMED	ACC	59.83±0.01	63.07±0.31	60.14±0.09	60.70±0.34	62.09±0.81	68.48±0.77	68.73±0.03	65.26±0.12	64.25±1.24	64.39±0.30	64.20±1.30	67.01±0.52	68.89±0.07	69.87±0.07
	NMI	31.05±0.02	26.32±0.57	22.44±0.14	23.67±0.29	23.84±3.54	30.61±1.71	28.26±0.03	24.80±0.17	23.88±1.05	26.67±1.31	22.87±2.04	31.59±1.45	31.43±0.13	32.20±0.08
	ARI	28.10±0.01	23.86±0.67	19.55±0.13	20.58±0.39	20.62±1.39	30.15±1.23	29.84±0.04	24.35±0.17	22.82±1.52	24.61±1.46	22.30±2.07	29.42±1.06	30.64±0.11	31.41±0.12
	F1	58.88±0.01	64.01±0.29	61.49±0.10	62.41±0.32	61.37±0.85	67.68±0.89	68.23±0.02	65.69±0.13	64.51±1.32	65.46±0.39	65.01±1.21	67.07±0.36	68.10±0.07	68.94±0.08
CORAFULL	ACC	26.27±1.10	33.12±0.19	31.92±0.45	32.19±0.31	29.60±0.81	32.66±1.29	34.35±1.00	22.07±0.43	29.57±0.59	29.75±0.69	26.67±0.40	31.52±2.95	37.51±0.81	38.80±0.60
	NMI	34.68±0.84	41.53±0.25	41.67±0.24	41.64±0.28	45.82±0.75	47.38±1.59	49.16±0.73	41.28±0.25	48.77±0.44	40.10±0.22	37.38±0.39	48.99±3.95	51.30±0.41	51.91±0.35
	ARI	9.35±0.57	18.13±0.27	16.98±0.29	17.17±0.22	17.84±0.86	20.01±1.38	22.60±0.47	12.38±0.24	18.80±0.57	16.47±0.38	13.63±0.27	19.11±2.63	24.46±0.48	25.25±0.49
	F1	22.57±1.09	28.40±0.30	27.71±0.58	27.72±0.41	25.95±0.75	29.06±1.15	26.96±1.33	18.85±0.41	25.43±0.62	24.62±0.53	22.14±0.43	26.51±2.87	31.22±0.87	31.68±0.76

Table 3: The average clustering performance with mean±std on six benchmarks. The red and blue values indicate the best and the runner-up results, respectively.

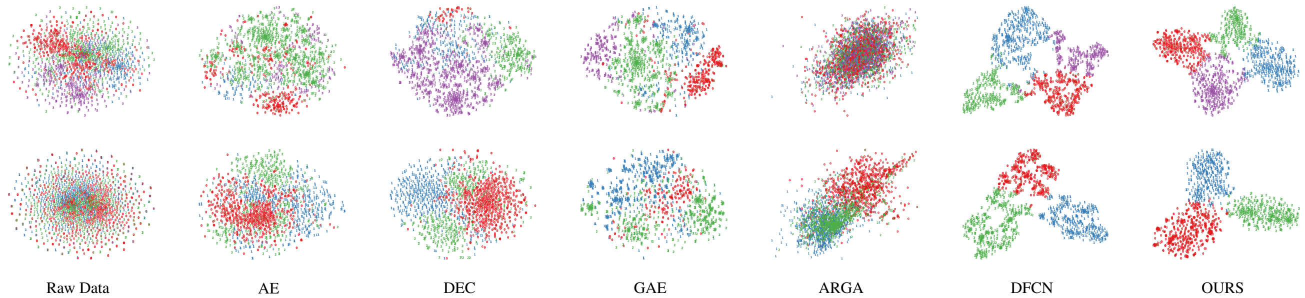


Figure 4: 2D visualization on two datasets. The first row and second row correspond to DBLP and ACM, respectively.

compared methods in terms of four metrics over all datasets. SDCN/SDCN$_Q$ (Bo et al. 2020), MVGRL (Hassani and Khasahmadi 2020) and DFCN (Tu et al. 2020) have been considered as three strongest deep clustering frameworks. Taking the results on DBLP for example, our DCRN exceeds DFCN by 3.66% 5.25%, 6.60% 3.58% increments with respect to ACC, NMI, ARI and F1. This is because both SDCN and DFCN overly introduce the attribute information learned by the auto-encoder part into the latent space, so that the node embedding contains redundant attributes about the sample, leading to representation collapse. In contrast, by reducing the information correlation in a dual manner, DCRN can learn more meaningful representation to improve the clustering performance; 2) it can be observed that the GCN-based clustering methods GAE/VGAE (Kipf and Welling 2016b), ARGA (Pan et al. 2019) and DAEGC (Wang et al. 2019) are not comparable with ours. This is because these methods do not consider to handle information correlation redundancy, thus resulting in the trivial constant representation; 3) our method improves the auto-encoder-based clustering methods, i.e., AE (Yang et al. 2017), DEC (Yang et al. 2017) and IDEC (Guo et al. 2017), by a large margin, all of which have been verified strong representation learning capacity for clustering on non-graph data, while these methods that merely rely on attribute information can not effectively learn discriminative information on graphs; 4) since K-means (Hartigan and Wong 1979) is directly performed on raw attributes, thus achieving unpromising results. Overall, the aforementioned observations have demonstrated the effectiveness of our proposed method in solving representation collapse issue. In the following section, ablation studies of each module in DCRN will be introduced in detail.

Ablation Studies

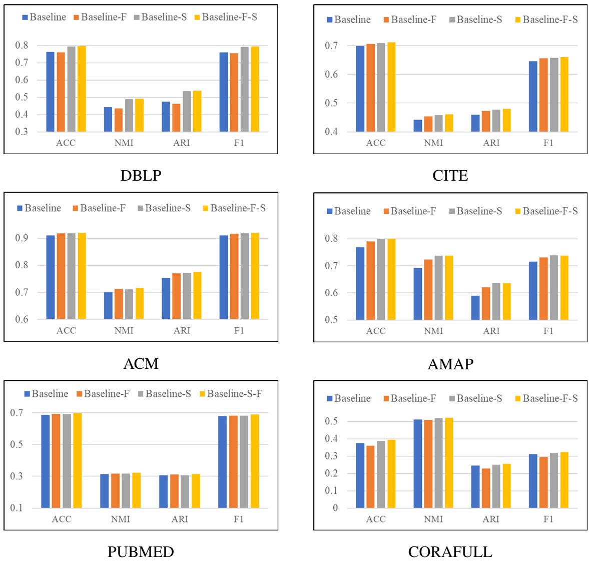
Effectiveness of DICR Module We conduct an ablation study to clearly verify the effectiveness of DICR module and report the results in Table 4. Here we denote the DFCN (Tu et al. 2020) as the Baseline since it's the feature extraction backbone of our network. Baseline-P, Baseline-D, and Baseline-P-D denote that the baseline adopts the propagated regularization, the DICR mechanism, and both. From the results in Table 4, we can observe that 1) compare with the baseline, Baseline-P has about 0.5% to 1.0% performance improvement in terms of four metrics on DBLP dataset. These results demonstrate that introducing a regularization term into the network training could improve the generalization capacity of the model as well as alleviate the over-smoothing; 2) Baseline-D consistently achieves better performance than that of the baseline. Taking the results on DBLP for example, Baseline-D exceeds the baseline by

Dataset	Metric	Baseline	Baseline-P	Baseline-D	Baseline-P-D
DBLP	ACC	76.00±0.80	77.00±0.41	79.63±0.27	79.66±0.25
	NMI	43.70±1.00	44.98±0.56	48.95±0.48	48.95±0.44
	ARI	47.00±1.50	48.51±0.84	53.48±0.51	53.60±0.46
	F1	75.70±0.80	76.77±0.38	79.26±0.28	79.28±0.26
CITE	ACC	69.50±0.20	70.07±0.21	70.88±0.19	70.86±0.18
	NMI	43.90±0.20	44.75±0.40	45.92±0.35	45.86±0.35
	ARI	45.50±0.30	46.52±0.36	47.73±0.29	47.64±0.30
	F1	64.30±0.20	65.03±0.23	65.79±0.20	65.83±0.21
ACM	ACC	90.90±0.20	91.57±0.12	91.91±0.21	91.93±0.20
	NMI	69.40±0.40	70.82±0.25	71.56±0.61	71.56±0.52
	ARI	74.90±0.40	76.68±0.28	77.50±0.53	77.56±0.52
	F1	90.80±0.20	91.53±0.12	91.90±0.21	91.94±0.20
AMAP	ACC	76.88±0.80	79.01±0.01	79.95±0.04	79.94±0.13
	NMI	69.21±1.00	72.29±0.01	73.69±0.05	73.70±0.24
	ARI	58.98±0.84	62.1±0.01	63.70±0.05	63.69±0.20
	F1	71.58±0.31	73.09±0.00	73.84±0.03	73.82±0.12
PUBMED	ACC	68.89±0.07	69.43±0.05	69.74±0.06	69.87±0.07
	NMI	31.43±0.13	31.98±0.12	32.04±0.06	32.20±0.08
	ARI	30.64±0.11	31.35±0.12	31.14±0.11	31.41±0.12
	F1	68.10±0.07	68.54±0.06	68.81±0.07	68.94±0.08
CORAFULL	ACC	37.51±0.81	37.04±0.71	38.23±0.59	38.80±0.60
	NMI	51.30±0.41	51.90±0.26	50.85±0.36	51.91±0.35
	ARI	24.46±0.48	24.13±0.51	24.83±0.37	25.25±0.49
	F1	31.22±0.87	30.35±0.87	31.34±0.81	31.68±0.76

Table 4: Ablation comparisons of DICR mechanism and the propagated regularization on six datasets.

3.63%, 5.25%, 6.48%, 3.56% performance increment with respect to ACC, NMI, ARI and F1. It benefits from that we conduct a DICR mechanism to enhance the discriminative capacity of the latent embedding for clustering performance improvement. We can obtain similar conclusions from the results on other datasets; 3) the results in the last column of Table 4 further verify the effectiveness of both components. As seen, Baseline-P-D achieves the best results compared to other variants.

Effectiveness of Dual Level Correlation Reduction To further investigate the superiority of the proposed DICR mechanism, we experimentally compare our method (i.e., Baseline-F-S in Fig. 5) with three counterparts. Likewise, we denote the DFCN as the Baseline. Baseline-F and Baseline-S are denoted that the Baseline merely adopts feature-level and sample-level correlation reduction strategy, respectively. From the results in Fig. 5, we can see that 1) Baseline-F outperforms Baseline in terms of four matrices on four of six datasets, but obtains unsatisfied performance on DBLP and CORAFULL. This is because the learned embedding is not robust without considering sample-level correlation redundancy; 2) the performance of Baseline-S is consistently better than that of Baseline over all datasets. For instance, Baseline-S obtains 3.60% accuracy improvement on DBLP. It shows that the decorrelation operation of samples is effective in filtering redundant information of two views while preserving more discriminative features for improving the clustering performance; 3) Baseline-F-S could leverage two types of correlation reduction to make the learned latent embedding more discriminative for better clustering. In summary, the above observations well demonstrate the effectiveness of dual level correlation reduction strategy.



DBLP

CITE

ACM

AMAP

PUBMED

CORAFULL

Figure 5: Ablation comparisons of dual information correlation reduction on six datasets.

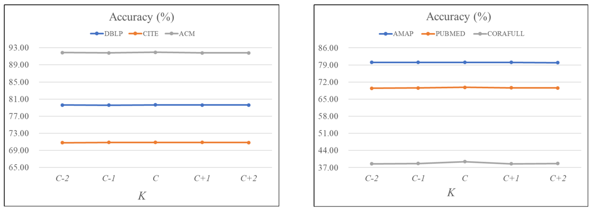


Figure 6: Clustering accuracy vs. hyper-parameter K.

Hyper-parameter Analysis of K Furthermore, we investigate the influence of hyper-parameter K. From Fig. 6, we observe that 1) the accuracy metric first increases to a high value and generally maintains it up to slight variation with the increasing value K; 2) the method tends to perform well when K is equal to the number of clusters C; 3) our DCRN is insensitive to the variation of the hyper-parameter K.

t-SNE Visualization of Clustering Results In order to show the superiority of DRCN intuitively, we visualize the distribution of the learned node embedding $\mathbf{Z}$ of DBLP and ACM generated by AE, DEC, GAE, ARGA, DFCN and our DCRN via t-SNE (Van der Maaten and Hinton 2008). As illustrated in Fig. 4, the visual results demonstrate that DCRN have a clearer structure, which can better reveal the intrinsic clustering structure among data.

Conclusion

In this work, we propose a novel self-supervised deep graph clustering network termed as Dual Correlation Reduction Network (DCRN). In our model, a carefully-designed dual information correlation reduction mechanism is introduced to reduce the information correlation in both sample and feature level. With this mechanism, the redundant information of the latent variables from two views can be filtered out and more discriminative features of both views can be well preserved. It plays an important role in avoiding representation collapse for better clustering. Experimental results on six benchmarks demonstrate the superiority of DCRN.

Acknowledgments

This work was supported by the National Key R&D Program of China (project no. 2020AAA0107100) and the National Natural Science Foundation of China (project no. 61922088, 61906020, 61872371 and 62006237).

References

Bo, D.; Wang, X.; Shi, C.; Zhu, M.; Lu, E.; and Cui, P. 2020. Structural deep clustering network. In *Proceedings of The Web Conference 2020*, 1400–1410.

Caron, M.; Bojanowski, P.; Joulin, A.; and Douze, M. 2018. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 132–149.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, 1597–1607. PMLR.

Chen, X.; and He, K. 2021. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15750–15758.

Fuglede, B.; and Topsoe, F. 2004. Jensen-Shannon divergence and Hilbert space embedding. In *International Symposium onInformation Theory, 2004. ISIT 2004. Proceedings.*, 31. IEEE.

Grill, J.-B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P. H.; Buchatskaya, E.; Doersch, C.; Pires, B. A.; Guo, Z. D.; Azar, M. G.; et al. 2020. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*.

Guo, X.; Gao, L.; Liu, X.; and Yin, J. 2017. Improved Deep Embedded Clustering with Local Structure Preservation. In *Ijcai*, 1753–1759.

Hartigan, J. A.; and Wong, M. A. 1979. Algorithm AS 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)*, 28(1): 100–108.

Hassani, K.; and Khasahmadi, A. H. 2020. Contrastive multi-view representation learning on graphs. In *International Conference on Machine Learning*, 4116–4126. PMLR.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9729–9738.

Horn, R. A. 1990. The hadamard product. In *Proc. Symp. Appl. Math*, volume 40, 87–169.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kipf, T. N.; and Welling, M. 2016a. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Kipf, T. N.; and Welling, M. 2016b. Variational graph autoencoders. *arXiv preprint arXiv:1611.07308*.

Kullback, S.; and Leibler, R. A. 1951. On information and sufficiency. *The annals of mathematical statistics*, 22(1): 79–86.

Liu, X.; Wang, L.; Zhu, X.; Li, M.; Zhu, E.; Liu, T.; Liu, L.; Dou, Y.; and Yin, J. 2019a. Absent multiple kernel learning algorithms. *IEEE transactions on pattern analysis and machine intelligence*, 42(6): 1303–1316.

Liu, X.; Zhu, X.; Li, M.; Wang, L.; Tang, C.; Yin, J.; Shen, D.; Wang, H.; and Gao, W. 2018. Late fusion incomplete multi-view clustering. *IEEE transactions on pattern analysis and machine intelligence*, 41(10): 2410–2423.

Liu, X.; Zhu, X.; Li, M.; Wang, L.; Zhu, E.; Liu, T.; Kloft, M.; Shen, D.; Yin, J.; and Gao, W. 2019b. Multiple kernel k k-means with incomplete kernels. *IEEE transactions on pattern analysis and machine intelligence*, 42(5): 1191–1204.

Page, L.; Brin, S.; Motwani, R.; and Winograd, T. 1999. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab.

Pan, S.; Hu, R.; Fung, S.-f.; Long, G.; Jiang, J.; and Zhang, C. 2019. Learning graph embedding with adversarial training methods. *IEEE transactions on cybernetics*, 50(6): 2475–2487.

Park, J.; Lee, M.; Chang, H. J.; Lee, K.; and Choi, J. Y. 2019. Symmetric graph convolutional autoencoder for unsupervised graph representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6519–6528.

Plummer, M. D.; and Lovász, L. 1986. *Matching theory*. Elsevier.

Shchur, O.; Mumme, M.; Bojchevski, A.; and Günnemann, S. 2018. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868*.

Tao, Z.; Liu, H.; Li, J.; Wang, Z.; and Fu, Y. 2019. Adversarial graph embedding for ensemble clustering. In *International Joint Conferences on Artificial Intelligence Organization*.

Tu, W.; Zhou, S.; Liu, X.; Guo, X.; Cai, Z.; Cheng, J.; et al. 2020. Deep Fusion Clustering Network. *arXiv preprint arXiv:2012.09600*.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).

Wang, C.; Pan, S.; Hu, R.; Long, G.; Jiang, J.; and Zhang, C. 2019. Attributed graph clustering: A deep attentional embedding approach. *arXiv preprint arXiv:1906.06532*.

Xie, J.; Girshick, R.; and Farhadi, A. 2016. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, 478–487. PMLR.

Yang, B.; Fu, X.; Sidiropoulos, N. D.; and Hong, M. 2017. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *international conference on machine learning*, 3861–3870. PMLR.

Yang, H.; Ma, K.; and Cheng, J. 2020. Rethinking graph regularization for graph neural networks. *arXiv preprint arXiv:2009.02027*.

You, Y.; Chen, T.; Sui, Y.; Chen, T.; Wang, Z.; and Shen, Y. 2020. Graph contrastive learning with augmentations. *Advances in Neural Information Processing Systems*, 33: 5812–5823.

Zbontar, J.; Jing, L.; Misra, I.; LeCun, Y.; and Deny, S. 2021. Barlow twins: Self-supervised learning via redundancy reduction. *arXiv preprint arXiv:2103.03230*.

Zhou, S.; Liu, X.; Li, M.; Zhu, E.; Liu, L.; Zhang, C.; and Yin, J. 2019. Multiple kernel clustering with neighbor-kernel subspace segmentation. *IEEE transactions on neural networks and learning systems*, 31(4): 1351–1362.

Zhou, S.; Zhu, E.; Liu, X.; Zheng, T.; Liu, Q.; Xia, J.; and Yin, J. 2020. Subspace segmentation-based robust multiple kernel clustering. *Information Fusion*, 53: 145–154.