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]

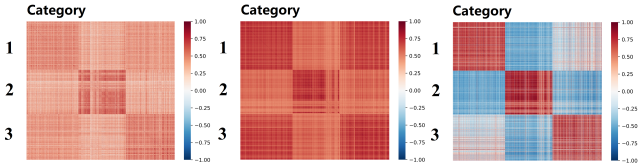
[1]College of Computer, National University of Defense Technology, Changsha, China
[2]College of Intelligence Science and Technology, National University of Defense Technology, Changsha, China
{yueliu, twx, xinwangliu, yangxihong, enzhu}@nudt.edu.cn, sihangjoe@gmail.com, slxnatavidad@163.com

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 a 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 dual level, thus improve 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 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 (papers, codes, datasets) 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,

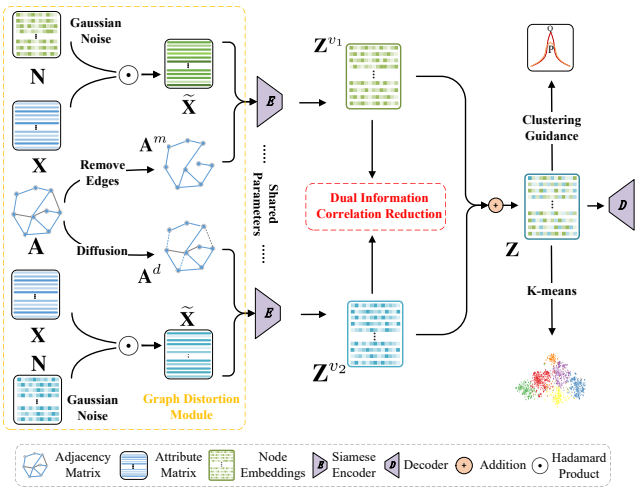


Figure 2: Illustration of the Dual Correlation Reduction Network (DCRN). In the proposed algorithm, the graph distortion module first generates two distorted graphs by introducing attribute and graph disturbances. Then, by forcing the same sample within two distorted graphs to have identical representations 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, $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ 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 $\mathbf{D} = diag(d_1, d_2, \ldots, d_N) \in \mathbb{R}^{N\times N}$ 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.

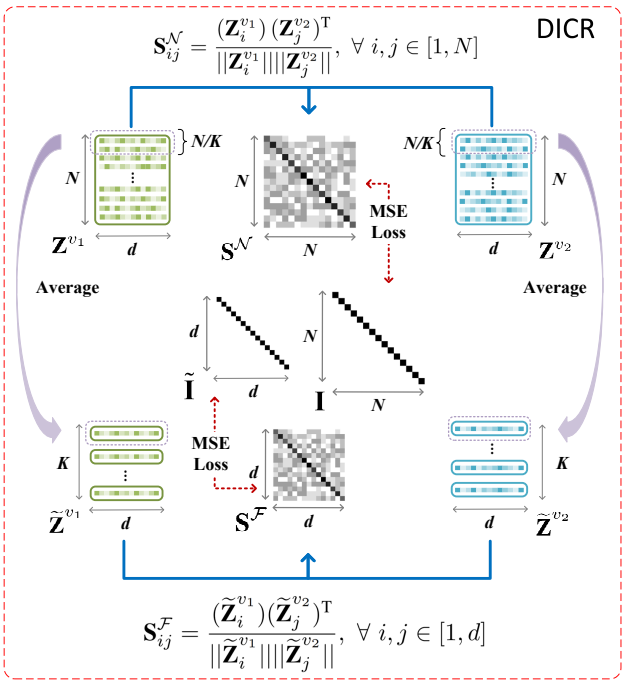


Figure 3: Illustration of Dual Information Correlation Reduction (DICR) mechanism.

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 γ and λ.
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. (5) and Eq. (8), respectively;
    Fuse $\mathbf{Z}^{v_1}$ and $\mathbf{Z}^{v_2}$ to obtain $\mathbf{Z}$ by Eq. (9);
    Calculate L_{DICR}, L_{REC}, and 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 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.

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

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.

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.

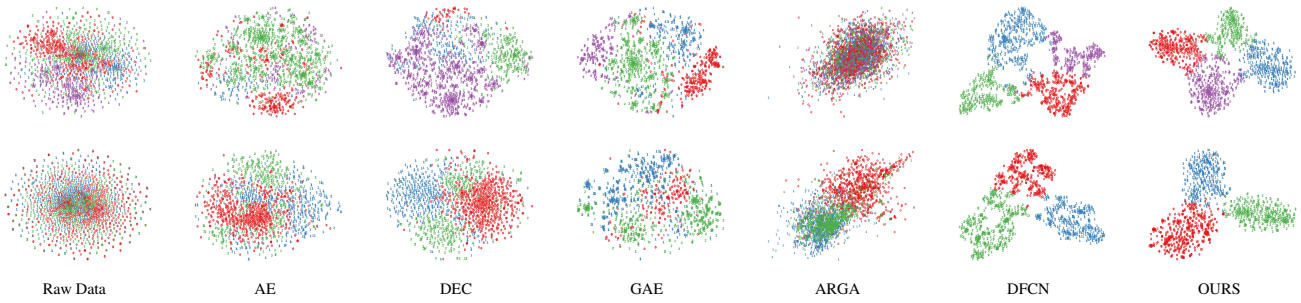


Figure 4: 2D visualization on two datasets. The first row and second row correspond to DBLP and ACM, respectively.

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.

Hyper-parameter Analysis of K Furthermore, we investigate the influence of hyper-parameters K. From Fig. 6, we

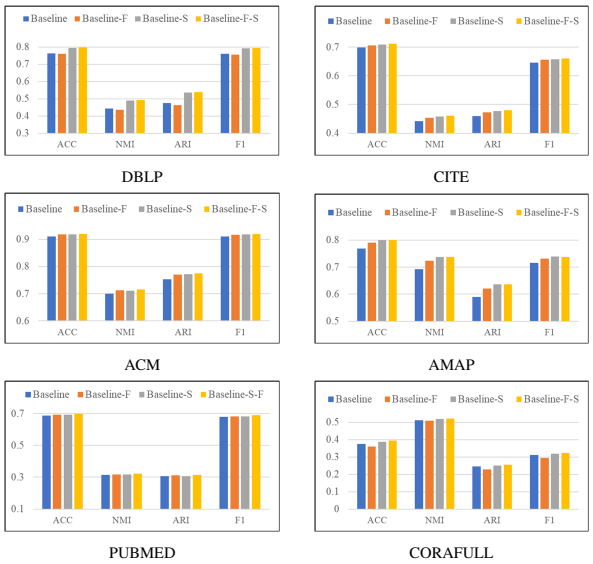


Figure 5: Ablation comparisons of dual information correlation reduction on six datasets.

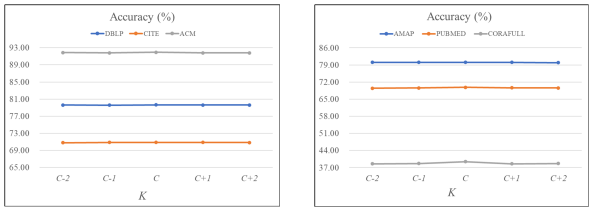


Figure 6: Clustering accuracy vs. hyper-parameter K.

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 vie 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 auto-encoders. *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.