Multi-level Graph Contrastive Prototypical Clustering

Yuchao Zhang[1,2] , **Yuan Yuan**[2] , **Qi Wang**[2] *
[1]School of Computer Science,
Northwestern Polytechnical University, Xi'an 710072, Shaanxi, P. R. China
[2]School of Artificial Intelligence, Optics and Electronics (iOPEN),
Northwestern Polytechnical University, Xi'an 710072, Shaanxi, P. R. China
yuchao_zhang@mail.nwpu.edu.cn, y.yuan1.ieee@gmail.com, crabwq@gmail.com

Abstract

Recently, graph neural networks (GNNs) have drawn a surge of investigations in deep graph clustering. Nevertheless, existing approaches predominantly are inclined to semantic-agnostic since GNNs exhibit inherent limitations in capturing global underlying semantic structures. Meanwhile, multiple objectives are imposed within one latent space, whereas representations from different granularities may presumably conflict with each other, yielding severe performance degradation for clustering. To this end, we propose a **M**ulti-**L**evel **G**raph **C**ontrastive **P**rototypical **C**lustering (MLG-CPC) framework for end-to-end clustering. Specifically, a **Pro**totype **Disc**rimination (ProDisc) objective function is proposed to explicitly capture semantic information via cluster assignments. Moreover, to alleviate the issue of objectives conflict, we introduce to perceive representations of different granularities within individual feature-, prototypical-, and cluster-level spaces by the feature decorrelation, prototype contrast, and cluster space consistency respectively. Extensive experiments on four benchmarks demonstrate the superiority of the proposed MLG-CPC against the state-of-the-art graph clustering approaches.

1 Introduction

As one of the most fundamental tasks in graph analysis, clustering divides nodes into different groups in absence of label annotations [Wang *et al.*, 2022b; Li *et al.*, 2020]. Recently, unsupervised graph representation learning based on GNNs [Kipf and Welling, 2017; Veličković *et al.*, 2018; Hamilton *et al.*, 2017] has provoked tremendous interest and shown promising capability for graph clustering [Park *et al.*, 2019; Zhu *et al.*, 2021]. In the existing literature, their approaches can be roughly divided into two categories, i.e., generative and contrastive graph clustering.

Generative methods prevalently resort to reconstruction objectives, which generate self-supervised information for representation learning. Following the auto-encoder paradigm, GAE [Kipf and Welling, 2016] first leverages to reconstruct the adjacency matrix. Inspired by adversarial mechanism, ARGA [Pan *et al.*, 2018] derives to refine the latent representations by discriminating real or fake samples. After that, DAEGC [Wang *et al.*, 2019] utilizes Kullback–Leibler divergence to simultaneously learn representations and find a better cluster-friendly space. Furthermore, DFCN [Tu *et al.*, 2021] considers multi-level features in a fusion manner. On the contrary, contrastive methods currently maximize agreements across views from an information theory perspective. DGI [Velickovic *et al.*, 2019] maximizes the mutual information (MI) between node representations and the global summary. MVGRL [Hassani and Khasahmadi, 2020] augments and contrasts with PageRank algorithm [Gasteiger *et al.*, 2018]. Inspired by Barlow Twins [Zbontar *et al.*, 2021], AGC-DRR [Gong *et al.*, 2022] reduces redundant information with an edge weight learner most recently.

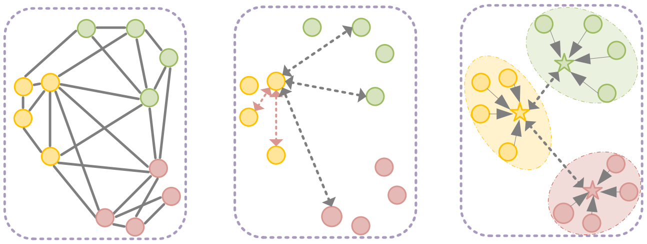


Figure 1: The middle picture shows the instance discrimination (dashed arrows) in graphs. We propose to generate prototypes and then conduct prototypical discrimination displayed in the right picture, alleviating the sampling bias occurred (red dashed arrows) in the middle picture. Colors indicate the class of nodes.

Despite empirical successes have witnessed advances of deep clustering, there exist following drawbacks to be addressed. (1) Many generative or contrastive methods [Kipf and Welling, 2016; Gong *et al.*, 2022] leverage shallow GNNs to avoid the over-smoothing and over-squashing phenomena [Keriven, 2022; Topping *et al.*, 2022]. However, semantic information as global knowledge is seldomly investigated, which thus theoretically is inclined to semantic-agnostic for graph clustering. Specifically, informative inner-class nodes in graphs generally scatter beyond one-hop neighbors, which benefits the downstream tasks. Generative methods based on reconstruction objectives excessively emphasize local neigh-

*Corresponding author.

bors by direct connections, which presumably even deteriorates the clustering performance. Contrastive methods also perceive limited global semantic information with shallow encoders and their objectives [Chen *et al.*, 2020] purely conjecture the rest of nodes are negatives, which simply pushes all the other nodes away, causing the sampling bias problem in Figure 1. (2) Some deep clustering approaches, such as SDCN [Bo *et al.*, 2020] and DFCN [Tu *et al.*, 2021] simultaneously consider multiple objectives in a fusion manner to explore multi-level information. Nevertheless, they impose objectives constraints within the same latent space regardless of whether representations have different granularities or not. This thus induces the conflict and presumably hinders optimization, leading serve performance degradation with laborious hyper-parameter searching and balancing.

In this paper, we introduce a novel Multi-Level Graph Contrastive Prototypical Clustering (MLG-CPC) to tackle aforementioned issues. Our goals include (1) optimizing representations of different granularities at multiple levels and (2) capturing semantic structures to explore the global information. To be specific, we first leverage encoders to distill feature representations from raw data, and then generate high-level semantic prototypes and clusters in sequence via projection heads as well as predictors on low-level yet fundamental feature representations. Concurrently, for representations of different granularities, we propose distinct optimization objectives within their respective spaces. Specifically, to facilitate the representation learning within high-level space, we constrain feature representations within feature-level space to be augmentation-invariant and non-degenerate. Then the distilled representations are transformed and mapped into prototypical-level space via projection heads. After that, prototypes are generated via cluster assignments for semantic exploitation by formulating a new Prototype Discrimination (ProDisc) loss function, which gathers and disperses prototypes between inter- and inner-views. Furthermore, we render output units of our cluster predictors to be identical with the amount of classes, generating the cluster assignments within cluster-level space. And the consistency of obtained cluster assignments across views can be constrained by NT-Xent [Chen *et al.*, 2020]. Overall, the features, prototypes, and clusters are individually optimized within their respective spaces, benefiting from each other and generating more promising representations for clustering. Our contributions are as follows:

- We propose a multi-level graph contrastive clustering method (MLG-CPC), which can simultaneously percieve representations of different granularities and conduct clustering in an end-to-end manner.
- To explore and exploit semantic knowledge, a Prototype Discrimination (ProDisc) objective function is derived on prototypes via cluster assignments.
- MLG-CPC optimizes different level of representations within individual spaces, avoiding laborious hyper-parameter searching to balance multiple objectives.
- Extensive experiments on four benchmarks demonstrate that the proposed MLG-CPC outperforms state-of-the-art graph clustering approaches.

2 Related Work

2.1 Deep Graph Clustering

Different from traditional graph clustering approaches such as probabilistic and matrix decomposition, deep graph clustering based on GNNs has received tremendous interest and advances by virtue of deep learning. Inspired by variation Bayes inference, GAE [Kipf and Welling, 2016] as the earliest deep clustering method leverages reconstruction objectives to reserve the graph structure. Following that, ARGA [Pan *et al.*, 2018] proposes to discriminate true or false samples from latent space with adversarial training. DAEGC [Wang *et al.*, 2019] simultaneously learns representations and finds a cluster-friendly space inspired by DEC [Xie *et al.*, 2016]. DFCN [Tu *et al.*, 2021] and SDCN [Bo *et al.*, 2020] both leverage more information from different views combined with DEC [Wang *et al.*, 2019], enhancing the quality of representations in a fusion manner. GALA [Park *et al.*, 2019] utilizes graph Laplacian sharpening to design a symmetric framework, further boosting the clustering performance. DGI [Velickovic *et al.*, 2019] adapts the idea of InfoMax into the graph domain, which maximizes the agreement between nodes and the global summary. MVGRL [Hassani and Khasahmadi, 2020] proposes to augment graphs with PageRank algorithm [Gasteiger *et al.*, 2018] and maximizes the MI across views based on multi-view learning. Most recently, AGC-DRR [Gong *et al.*, 2022] proposes an adversarial edge leaner, which reduces redundant information across views benefitted from Barlow Twins [Zbontar *et al.*, 2021]. Albeit above approaches ameliorate graph clustering from various perspectives, methods such as DFCN and SDCN optimize different objectives and then fuse within the same space. Nevertheless, representations of different granularities may conflict with each other, thus requiring laborious parameters searching and balancing. Meanwhile, most approaches do not take global semantic structures into consideration, which is harmful for graph clustering.

2.2 Semantic Structures in Contrastive Learning

Prototypes can model the class semantic information, which are widely applied to few-shot learning [Snell *et al.*, 2017]. Different from limited yet available labels in few-shot learning, there is no annotations in unsupervised scenarios. PCL [Li *et al.*, 2021] proposes prototypical contrastive learning for images with Expectation-Maximization algorithms, which performs node-prototype interaction to improve the quality of image representations. However, this is theoretically non-trivial for graphs due to their assumptions, where images follow the independently and identically distributed (IID) hypothesis [Yuan *et al.*, 2018; Liu *et al.*, 2022; Wang *et al.*, 2022a] while nodes in graphs manifest strong intrinsic dependences by edges. Similarly, SwAV [Caron *et al.*, 2020] simultaneously learns representations and cluster assignments by extending the idea of deepCluster [Caron *et al.*, 2018], while they still omit multi-level information. For graphs semantic exploitation, GraphLog [Xu *et al.*, 2021] generates hierarchical prototypes and performs instance-prototype alignment. PGCL [Lin *et al.*, 2022] extends the SwAV for graph domain, improving the representations of molecules.

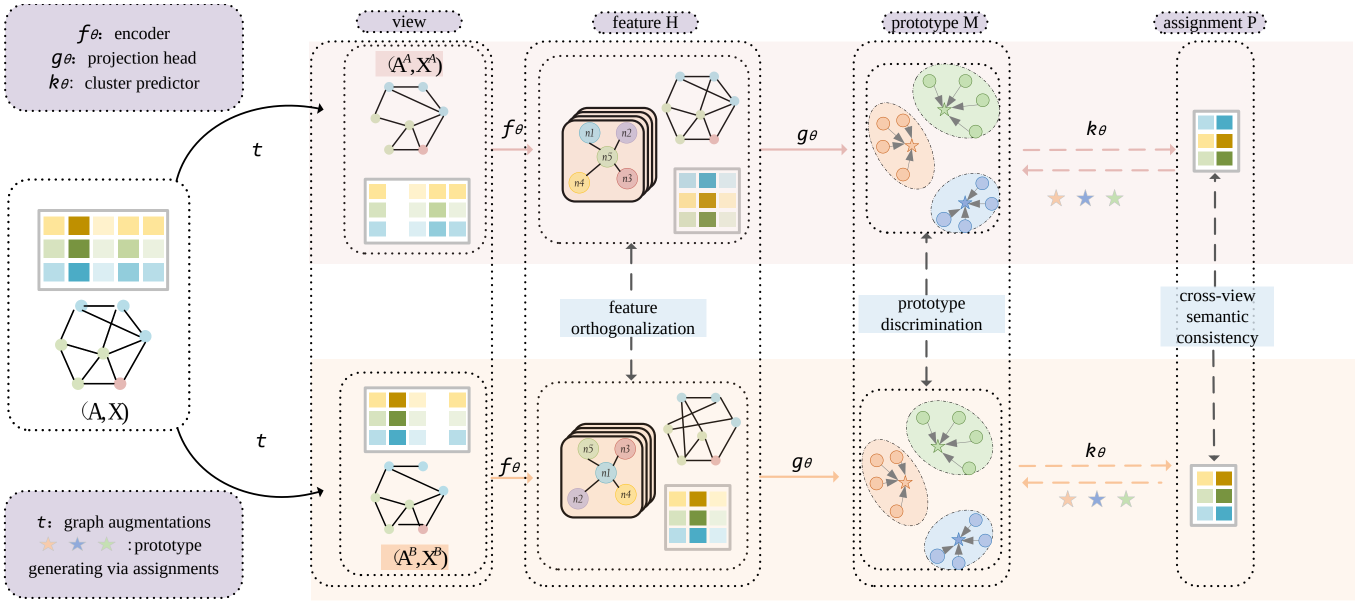


Figure 2: Overview of the proposed MLG-CPC. First, we generate two views by graph augmentations. Secondly, we learn representations of different granularities, i.e., $\mathbf{H}$ via encoders f_θ, $\mathbf{Z}$ and $\mathbf{M}$ via projection heads g_θ, as well as $\mathbf{P}$ and $\mathbf{Q}$ via cluster predictors k_θ respectively. Then three distinct losses are formulated within their respective spaces. After that, the overall objective function is optimized jointly.

Notations	Descriptions
$\mathbf{A} \in \mathbb{R}^{N\times N}$	Original adjacency matrix
$\mathbf{X} \in \mathbb{R}^{N\times D}$	Original feature matrix
$\mathbf{A}^v \in \mathbb{R}^{N\times N}$	Augmented adjacency matrix in v-th view
$\mathbf{X}^v \in \mathbb{R}^{N\times D}$	Augmented feature matrix in v-th view
$\mathbf{H}^v \in \mathbb{R}^{N\times d}$	Feature representation in v-th view
$\mathbf{Z}^v \in \mathbb{R}^{N\times d}$	Projected $\mathbf{H}^v$ in prototypical-level space
$\mathbf{M}^v \in \mathbb{R}^{K\times d}$	Prototype representation in v-th view
$\mathbf{P}^v \in \mathbb{R}^{N\times K}$	Cluster assignment matrix in v-th view
$\mathbf{Q}^v \in \mathbb{R}^{K\times N}$	Cluster statistic matrix in v-th view

Table 1: Notations for MLG-CPC

Nonetheless, these approaches are devised for graph-level and are task-agnostic. Our MLG-CPC is for end-to-end clustering and posses different motivations with distinct objectives. Moreover, these methods adopt cosine similarities to compute the cluster probabilities whereas our method can directly obtain the class assignments by virtue of end-to-end architecture and cluster space, which may shed light upon multi-view graph clustering for further investigations.

3 Methodology

Given a graph, we initially augment this original graph with graph augmentations to produce two different views. Then the augmented graphs are first transformed into $\mathbf{H}$ via message passing encoders f_θ, which are then projected by the projection head g_θ to generate $\mathbf{Z}$ and construct prototypes $\mathbf{M}$. Finally, we leverage cluster predictors k_θ to learn assignments $\mathbf{P}$ and statistic matrix $\mathbf{Q}$. Three distinct losses are formulated and optimized at different levels on $\mathbf{H}$, $\mathbf{M}$, $\mathbf{P}$, and $\mathbf{Q}$ with respective feature-, prototypical-, and cluster-level spaces. The framework of our MLG-CPC is illustrated in Figure 2.

3.1 Notations and Problem Definitions

Let an attributed graph $\mathbf{G} = \{\mathbf{V}, \mathbf{A}, \mathbf{X}\}$, where nodes set $\mathbf{V} = \{\mathbf{v_1}, \mathbf{v_2}, \dots, \mathbf{v_N}\}$ associated with edges. The structure of G can be defined by an adjacency matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$. Nodes with features can be denoted as $\mathbf{X} \in \mathbb{R}^{N\times D}$. $\mathbf{I} \in \mathbb{R}^{D\times D}$ is the identity matrix. And lower case of the matrix denotes vectors. The goal of graph clustering is to separate the nodes into different clusters without requiring any annotations. The concrete notations are encapsulated in Table 1.

3.2 Message Passing and Graph Augmentations

Encoders Based on Message Passing. Following the message passing neural network (MPNN) [Gilmer *et al.*, 2017], we update the node by transporting messages of it and its neighbors, which can be defined as:

$$\mathbf{h}^k_{\mathcal{N}(v)} \leftarrow \mathrm{MESSAGE}_k\left(\left\{\mathbf{h}^{k-1}_u, \forall u \in \mathcal{N}(v)\right\}\right), \quad (1)$$

$$\mathbf{h}^k_v \leftarrow \sigma\left(\mathbf{W}^k \cdot \mathrm{UPDATES}\left(\mathbf{h}^{k-1}_v, \mathbf{h}^k_{\mathcal{N}(v)}\right)\right), \quad (2)$$

where $\mathcal{N}(v)$ denotes the neighbors representations of node v. $\mathrm{MESSAGE}_k$ is the message function at layer k, which should be permutation invariant. $\mathbf{h}^k_{\mathcal{N}(v)} \in \mathbb{R}^{1\times d}$ denotes messages of neighbors gathered by node v. UPDATES operator updates the node. $\mathbf{W}^k \in \mathbb{R}^{D\times d}$ is the transformation matrix at layer k. $\mathbf{h}^k_v$ is the updated node v at layer k and $h^0_v = x_v$. For simplicity, we set the $\mathrm{MESSAGE}_k$ and UPDATES to be mean operation. Then the encoder can be represented as:

$$\mathbf{h}^k_v \leftarrow \sigma\left(\mathbf{W} \cdot \mathrm{MEAN}\left(\left\{\mathbf{h}^{k-1}_v\right\} \cup \left\{\mathbf{h}^k_{\mathcal{N}(v)}\right\}\right)\right). \quad (3)$$

By this means, we locally generate feature representations $\mathbf{H}$ into feature-level space from raw data $\mathbf{X}$ and $\mathbf{A}$. For global semantic exploration, we will detail it in section 3.3.

Graph Augmentations. Following the multi-view contrastive learning paradigm, we resort to contrasting across views and adopt two widely used graph augmentations techniques, edge dropping and feature masking, which augment graphs from perspectives of structures and features. Specifically, for edge dropping, we randomly drop a portion of edges in graphs by a mask matrix $\mathbf{R} \in \{0,1\}^{N \times N}$, whose elements are produced via Bernoulli distribution with probability r_i for generating view $i \in \{A, B\}$, which can be defined as:

$$\mathbf{A}^i = \mathbf{A} \circ \mathbf{R}, \tag{4}$$

where $\circ$ is Hadamard product. For feature masking, we define an indicator $\mathbf{m} \in \{0,1\}^{d \times 1}$ with probability m_i for assigning 0 to features, and then mask a portion of the node features for generating view $i \in \{A, B\}$, which can be computed as:

$$\mathbf{X}^i = \left[\mathbf{x}_1 \circ \mathbf{m}; \mathbf{x}_2 \circ \mathbf{m}; \cdots; \mathbf{x}_N \circ \mathbf{m}\right]^\top . \tag{5}$$

Note that different from previous approaches [Xia *et al.*, 2022], which set distinct value for two views, we set identical values for two views to draw them from the same distribution.

3.3 Multi-Level Graph Clustering

Feature Orthogonality with Invariance. Given the representations $\mathbf{H}$ of a graph, we formulate the feature distillation problem within feature-level space inspired by W-MSE [Ermolov *et al.*, 2021], which is represented as:

$$\begin{gathered}\min_\theta \mathbb{E}\left[\operatorname{dist}\left(\mathbf{H}^A_{\cdot,i}, \mathbf{H}^B_{\cdot,i}\right)\right], \\ \text{s.t. } \operatorname{cov}\left(\mathbf{H}^A, \mathbf{H}^A\right) = \operatorname{cov}\left(\mathbf{H}^B, \mathbf{H}^B\right) = \mathbf{I},\end{gathered} \tag{6}$$

where dist denotes the mean squared error, i.e., the L2 distance, and cov is the covariance measurement between variables across views. Different from W-MSE whitens the embedding for implicitly satisfying the condition, we tackle this problem by via Lagrangian multiplier, so $\mathcal{L}_{fl}$ is computed as:

$$\begin{gathered}\min_\theta \mathbb{E}\left[\operatorname{dist}\left(\mathbf{H}^A_{\cdot,i}, \mathbf{H}^B_{\cdot,i}\right)\right] \\ +\lambda \cdot \left(\left\|\left(\mathbf{H}^A\right)^\top \mathbf{H}^A - \mathbf{I}\right\|^2 + \left\|\left(\mathbf{H}^B\right)^\top \mathbf{H}^B - \mathbf{I}\right\|^2\right),\end{gathered} \tag{7}$$

where the first term renders representations across views to be invariant to augmentations, capturing the permutation invariant information. The second term encourages different features in the same view to be orthogonal, minimizing correlation of features within views. λ is a parameter to balance the objective and conditions.

Prototypical-level Contrast. SimCLR [Chen *et al.*, 2020] adopts instance discrimination strategy, which is semantic-agnostic. We explore and regard prototypes as globally semantic structures and propose to conduct discrimination on prototypical-level. Specifically, we adopt Multi-Layer Perceptions (MLPs) as projection heads g_θ within prototypical-level space. The prototypes constructed by cluster assignments generated by cluster predictors, formally, we calculate prototypes for each cluster as follows:

$$\boldsymbol{\mu}_k = \frac{\sum_{\mathbf{z}} p(k \mid \mathbf{z}) \cdot \mathbf{z}}{\left\|\sum_{\mathbf{z}} p(k \mid \mathbf{z}) \cdot \mathbf{z}\right\|_2}, \tag{8}$$

where $\mathbf{z} \in \mathbb{R}^{1 \times d}$ is the output by projection heads based on h, and $p(k \mid \mathbf{z})$ is the cluster assignment generated by predictors, which will be detailed in section 3.3.3. Cluster assignments indicate the node belong to certain cluster k. Then we define our ProDisc loss for one pair as:

$$\ell\left(\boldsymbol{\mu}_i^A, \boldsymbol{\mu}_i^B\right) = -\log \frac{e^{\theta\left(\boldsymbol{\mu}_i^A, \boldsymbol{\mu}_i^B\right)/\tau}}{\Phi_{\text{inter}} + \Phi_{\text{intra}}}, \tag{9}$$

where $\theta\left(,\right)/\tau$ is the cosine similarity with temperature parameter. Φ sums total negative sample pairs from inter or intra perspectives and K is the number of classes. Thus, the denominator can disperse negative prototypes pairs from different views and the same views:

$$\Phi_{\text{inter}} = \sum_{k=1}^{K} e^{\theta\left(\boldsymbol{\mu}_i^A, \boldsymbol{\mu}_k^B\right)/\tau}, \tag{10}$$

$$\Phi_{\text{intra}} = \sum_{k=1, k \neq i}^{K} e^{\theta\left(\boldsymbol{\mu}_i^A, \boldsymbol{\mu}_k^A\right)/\tau}, \tag{11}$$

Then we calculate the loss for all pairs symmetrically, so the overall ProDisc loss of prototypical-level is:

$$\mathcal{L}_{pl} = \frac{1}{2N} \sum_{i=1}^{N} \left[\ell\left(\boldsymbol{\mu}_i^A, \boldsymbol{\mu}_i^B\right) + \ell\left(\boldsymbol{\mu}_i^B, \boldsymbol{\mu}_i^A\right)\right]. \tag{12}$$

The prototype discrimination interacts globally from the semantic perspective, alleviating sampling bias to some extent. Moreover, similar to contrastive algorithms, this loss is theoretically coherent with [Wang and Isola, 2020], where the positive prototype pair encourages the alignment and negative prototype pairs guarantee the uniformity.

Cluster Space Consistency. Consequently, we formulate our cluster predictors k_θ via MLPs followed by softmax activation. The units of predictor are equal to the amount of classes K. By this way, our model can predict nodes cluster assignments $\mathbf{P}^i \in \mathbb{R}^{N \times K}$ of two views based on $\mathbf{Z}^i$. To constrain the consistency of two cluster distributions, we adopt the idea of cross-entropy and leverage a more flexible variant of it [Chen *et al.*, 2020], which is defined as:

$$\ell\left(\mathbf{p}_i^A, \mathbf{p}_i^B\right) = -\log \frac{e^{\theta\left(\mathbf{p}_i^A, \mathbf{p}_i^B\right)/\tau}}{\sum_{k=1, k \neq i}^{N} e^{\theta\left(\mathbf{p}_i^A, \mathbf{p}_k^B\right)/\tau}}, \tag{13}$$

we calculate the loss for all pairs symmetrically, so the assignments consistency loss is:

$$\mathcal{L}_{acl} = \frac{1}{2N} \sum_{i=1}^{N} \left[\ell\left(\mathbf{p}_i^A, \mathbf{p}_i^B\right) + \ell\left(\mathbf{p}_i^B, \mathbf{p}_i^A\right)\right]. \tag{14}$$

In addition, we regard column $\mathbf{p}_i$ of $\mathbf{P}$ as the cluster statistics vectors $\mathbf{q}_i$, and further constrain the consistency between them for stability, which is denoted as:

$$\ell\left(\mathbf{q}_i^A, \mathbf{q}_i^B\right) = -\log \frac{e^{\theta\left(\mathbf{q}_i^A, \mathbf{q}_i^B\right)/\tau}}{\sum_{k=1, k \neq i}^{K} e^{\theta\left(\mathbf{q}_i^A, \mathbf{q}_k^B\right)/\tau}}, \tag{15}$$

Algorithm 1 The training procedure of MLG-CPC

Input: Graph $\mathbf{G} = \{\mathbf{V}, \mathbf{A}, \mathbf{X}\}$; The number of clusters K; Maximum iterations T; Edge dropping rates r_i and feather masking rates m_i; Hyper-parameter λ.
Output:The clustering results.

```
for $i = 1$ to $T$ do
    Obtain $G^B$ via Eq.(4) and Eq.(5) augmentations.
    Calculate feature $H^A$ and $H^B$ via Eq.(3).
    Calculate representations $\mu^A$ and $\mu^B$ via Eq.(8).
    Calculate assignments $p^A$ and $p^B$ by predictor.
    Calculate statistics $q^A$ and $q^B$ by transposition.
    Calculate $\mathcal{L}_{fl}$, $\mathcal{L}_{pl}$, and $\mathcal{L}_{cl}$, respectively.
    Update MLG-CPC via minimizing Eq.(18).
end for
Calculate clustering results via the clustering assignments $P_1$ and $P_2$.
return The clustering results.
```

and we calculate the loss for all pairs symmetrically, so the statistic consistency loss is:

$$\mathcal{L}_{scl} = \frac{1}{2N} \sum_{i=1}^{N} \left[\ell\left(\mathbf{q}_i^A, \mathbf{q}_i^B\right) + \ell\left(\mathbf{q}_i^B, \mathbf{q}_i^A\right) \right]. \qquad (16)$$

Here, the overall cluster-level consistency loss is:

$$\mathcal{L}_{cl} = \mathcal{L}_{acl} + \mathcal{L}_{scl}. \qquad (17)$$

3.4 Objective Function

We separately formulate different levels of representations within their own space, avoiding the objectives conflict issue. We do not need specific hyper-parameter tuning to balance the different objectives [Liu *et al.*, 2023], thus keeping importance of each objective identical. The total objective function of our MLG-CPC is denoted as:

$$\mathcal{L} = \mathcal{L}_{fl} + \mathcal{L}_{pl} + \mathcal{L}_{cl}. \qquad (18)$$

Feature representations are distilled by feature orthogonality in feature-level space, and then contribute to generating accurate prototypes within prototypical-level space. Consequently, representations refined by prototype discrimination are further optimized via cluster-level consistency. Lastly, we average cluster assignments from two views as labels output by our MLG-CPC for evaluation testing. The procedure of MLG-CPC is illustrated in Algorithm 1.

4 Experiments

In following sections, we first introduce experimental setups. Second, we conduct extensive experiments of graph clustering and verify the effectiveness of different sub-modules. After that, we investigate the influence of different graph augmentation strategies. Finally, the learned representations and similarity matrices are visualized for display intuitively.

4.1 Datasets

We adopt four commonly used graph datasets [Shchur *et al.*, 2018] in our experiments including CITE, ACM, DBLP, and AMAP. Each dataset contains an adjacency matrix and a feature matrix. Concrete descriptions of these datasets are illustrated in Table 2.

Dataset	#Node	#Dimension	#Edges	#Class
ACM	3025	1870	13128	3
DBLP	4057	334	3528	4
CITE	3327	3703	4552	6
AMAP	7650	745	119081	8

Table 2: Datasets in experiments

4.2 Experiment Setup

Implementation Details. We implement our MLG-CPC on PyTorch platform and Deep Graph Library (DGL)[1] with the NVIDIA GeForce RTX 3090. Our encoders and prototype projection heads are shared graph convolutional networks (GCNs) and MLPs, respectively. Also, we leverage one-layer MLP followed by softmax to construct cluster predictors.

Parameters Settings. For all datasets, we set the learning rate at 1e-3, τ at 0.2, and λ at 5e-4. We use grid-search to find the optimal graph augmentation parameters, i.e., edge dropping and feature masking rates ranging from 0 to 1. We will analyse these two parameters in the following section. The training procedure of MLG-CPC is optimized until reaching max epochs. We set max epochs at 40, 200, 400, 1000 for CITE, AMAP, ACM, and DBLP, respectively. We report the average mean scores and standard deviation of 10 times running. For other baselines, we follow and directly compare the results reported in AGC-DRR [Gong *et al.*, 2022].

Evaluation Metrics. In our graph clustering experiments, the Accuracy (ACC), Normalized Mutual Information (NMI), adjusted rand index (ARI), and F1-score (F1) [Tu *et al.*, 2021; Bo *et al.*, 2020] are presented for evaluation.

4.3 Clustering Results

Comparison Methods To demonstrate the validity of our MLG-CPC, we compare with 12 clustering methods. K-means [Hartigan and Wong, 1979] as one of the most classic algorithm can cluster data into different groups. AE [Yang *et al.*, 2017] follows the auto-encoder paradigm to perform clustering. DEC [Xie *et al.*, 2016] and IDEC [Guo *et al.*, 2017] further enhance the AE with co-clustering mechanism, GAE/VGAE [Kipf and Welling, 2016] and DAEGC [Wang *et al.*, 2019] extends the idea of AE to the graph domain. ARGA [Pan *et al.*, 2018] combines adversarial networks with GAE. SDCN [Bo *et al.*, 2020] and DFCN [Tu *et al.*, 2021] integrate the representations of AE and GAE in a fusion manner. MVGRL [Hassani and Khasahmadi, 2020] adopts the MI maximization and Pagerank to improve the quality of representations. AGC-DRR [Gong *et al.*, 2022] devises a edge learner for more robust clustering.

We present the graph clustering results in Table 3. Compared with the first three methods, our MLG-CPC consistently outperforms them by a large margin. These methods do not utilize the graph topological information though they are effective for tackling non-graph data such as images and tabular samples. Compared with generative graph clustering algorithms such as GAE [Kipf and Welling, 2016], ARGA [Pan

[1] https://www.dgl.ai/

Method	ACM				DBLP			
	ACC	NMI	ARI	F1	ACC	NMI	ARI	F1
K-means	67.31 ± 0.71	32.44 ± 0.46	30.60 ± 0.69	67.57 ± 0.74	38.65 ± 0.65	11.45 ± 0.38	6.97 ± 0.39	31.92 ± 0.27
AE	81.83 ± 0.08	49.30 ± 0.16	54.64 ± 0.16	82.01 ± 0.08	51.43 ± 0.35	25.40 ± 0.16	12.21 ± 0.43	52.53 ± 0.36
DEC	84.33 ± 0.76	54.54 ± 1.51	60.64 ± 1.87	84.51 ± 0.74	58.16 ± 0.56	29.51 ± 0.28	23.92 ± 0.39	59.38 ± 0.51
IDEC	85.12 ± 0.52	56.61 ± 1.16	62.16 ± 1.50	85.11 ± 0.48	60.31 ± 0.62	31.17 ± 0.50	25.37 ± 0.60	61.33 ± 0.56
GAE	84.52 ± 1.44	55.38 ± 1.92	59.46 ± 3.10	84.65 ± 1.33	61.21 ± 1.22	30.80 ± 0.91	22.02 ± 1.40	61.41 ± 2.23
VGAE	84.13 ± 0.22	53.20 ± 0.52	57.72 ± 0.67	84.17 ± 0.23	58.59 ± 0.06	26.92 ± 0.06	17.92 ± 0.07	58.69 ± 0.07
DAEGC	86.94 ± 2.83	56.18 ± 4.15	59.35 ± 3.89	87.07 ± 2.79	62.05 ± 0.48	32.49 ± 0.45	21.03 ± 0.52	61.75 ± 0.67
ARGA	86.29 ± 0.36	56.21 ± 0.82	63.37 ± 0.86	86.31 ± 0.35	64.83 ± 0.59	29.42 ± 0.92	27.99 ± 0.91	64.97 ± 0.66
ARVGA	83.89 ± 0.54	51.88 ± 1.04	57.77 ± 1.17	83.87 ± 0.55	54.41 ± 0.42	25.90 ± 0.33	19.81 ± 0.42	55.37 ± 0.40
SDCN_Q	86.95 ± 0.08	58.90 ± 0.17	65.25 ± 0.19	86.84 ± 0.09	65.74 ± 1.34	35.11 ± 1.05	34.00 ± 1.76	65.78 ± 1.22
SDCN	90.45 ± 0.18	68.31 ± 0.25	73.91 ± 0.40	90.42 ± 0.19	68.05 ± 1.81	39.50 ± 1.34	39.15 ± 2.01	67.71 ± 1.51
MVGRL	86.73 ± 0.76	60.87 ± 1.40	65.07 ± 1.76	86.85 ± 0.72	42.73 ± 1.02	15.41 ± 0.63	8.22 ± 0.50	40.52 ± 1.51
DFCN	90.90 ± 0.20	69.40 ± 0.40	74.90 ± 0.40	90.80 ± 0.20	76.00 ± 0.80	43.70 ± 1.00	47.00 ± 1.50	75.70 ± 0.80
AGC-DRR	92.55 ± 0.09	72.89 ± 0.24	79.08 ± 0.24	92.55 ± 0.09	80.41 ± 0.47	49.77 ± 0.65	55.39 ± 0.88	79.90 ± 0.45
Ours	**93.20 ± 0.12**	**75.57 ± 0.10**	**81.11 ± 0.10**	**93.16 ± 0.14**	**82.13 ± 0.52**	**52.41 ± 0.61**	**57.78 ± 0.82**	**80.32 ± 0.47**

Method	CITE				AMAP			
	ACC	NMI	ARI	F1	ACC	NMI	ARI	F1
K-means	39.32 ± 3.17	16.94 ± 3.22	13.43 ± 3.02	36.08 ± 3.53	27.22 ± 0.76	13.23 ± 1.33	5.50 ± 0.44	23.96 ± 0.51
AE	57.08 ± 0.13	27.64 ± 0.08	29.31 ± 0.14	53.80 ± 0.11	48.25 ± 0.08	38.76 ± 0.30	20.80 ± 0.47	47.87 ± 0.20
DEC	55.89 ± 0.20	28.34 ± 0.30	28.12 ± 0.36	52.62 ± 0.17	47.22 ± 0.08	37.35 ± 0.05	18.59 ± 0.04	46.71±0.12
IDEC	60.49 ± 1.42	27.17 ± 2.40	25.70 ± 2.65	61.62 ± 1.39	47.62 ± 0.08	37.83 ± 0.08	19.24 ± 0.07	47.20 ± 0.11
GAE	61.35 ± 0.80	34.63 ± 0.65	33.55 ± 1.18	57.36 ± 0.82	71.57 ± 2.48	62.13 ± 2.79	48.82 ± 4.57	68.08 ± 1.76
VGAE	60.97 ± 0.36	32.69 ± 0.27	33.13 ± 0.53	57.70 ± 0.49	74.26 ± 3.63	66.01 ± 3.40	56.24 ± 4.66	70.38 ± 2.98
DAEGC	64.54 ± 1.39	36.41 ± 0.86	37.78 ± 1.24	62.20 ± 1.32	76.44 ± 0.01	65.57 ± 0.03	59.39 ± 0.02	69.97 ± 0.02
ARGA	61.07 ± 0.49	34.40 ± 0.71	34.32 ± 0.70	58.23 ± 0.31	69.28 ± 2.30	58.36 ± 2.76	44.18 ± 4.41	64.30 ± 1.95
ARVGA	59.31 ± 1.38	31.80 ± 0.81	31.28 ± 1.22	56.05 ± 1.13	61.46 ± 2.71	53.25 ± 1.91	38.44 ± 4.69	58.50 ± 1.70
SDCN_Q	61.67 ± 1.05	34.39 ± 1.22	35.50 ± 1.49	57.82 ± 0.98	35.53 ± 0.39	27.90 ± 0.40	15.27 ± 0.37	34.25 ± 0.44
SDCN	65.96 ± 0.31	38.71 ± 0.32	40.17 ± 0.43	63.62 ± 0.24	53.44 ± 0.81	44.85 ± 0.83	31.21 ± 1.23	50.66 ± 1.49
MVGRL	68.66 ± 0.36	43.66 ± 0.40	44.27 ± 0.73	63.71 ± 0.39	45.19 ± 1.79	36.89 ± 1.31	18.79 ± 0.47	39.65 ± 2.39
DFCN	**69.50 ± 0.20**	43.90 ± 0.20	45.50 ± 0.30	64.30 ± 0.20	76.88 ± 0.80	69.21 ± 1.00	58.98 ± 0.84	71.58 ± 0.31
AGC-DRR	68.32 ± 1.83	43.28 ± 1.41	45.34 ± 2.33	**64.82 ± 1.60**	78.11 ± 1.69	**72.21 ± 1.63**	61.15 ± 1.65	72.72 ± 0.97
Ours	69.31 ± 1.47	**44.47 ± 0.58**	**45.66 ± 0.69**	64.21 ± 1.25	**79.65 ± 0.94**	68.57 ± 1.32	**64.13 ± 1.21**	**75.52 ± 0.76**

Table 3: The average clustering performance with mean±std on four benchmarks. The bold and underlined values indicate the best and the second best results, respectively.

et al., 2018], DAEGC [Wang *et al.*, 2019], which pursue reconstruction for optimization. From the table, we can see MLG-CPC outperforms DAEGC [Wang *et al.*, 2019], a state-of-the-art generative graph clustering algorithm, by 6.26%, 20.08%, 4.77%, 3.21% on ACM, DBLP, CITE, and AMAP datasets in terms of ACC evaluation, respectively. Meanwhile, compared with contrastive and fusion-fashion graph clustering methods such as MVGRL [Hassani and Khasahmadi, 2020], AGC-DRR [Gong *et al.*, 2022], and DFCN [Tu *et al.*, 2021], our method perceives multi-level optimization, and different granularities of representations facilitate each other to some degree. Moreover, MLG-CPC takes global semantic structures into consideration and optimizes different objectives with distinct spaces, whereas these methods still omits the semantic knowledge and might be risk of objectives optimization conflict issues. Overall, aforementioned clustering performance observations have demonstrated the effectiveness of our MLG-CPC for graph clustering.

4.4 Ablation Study

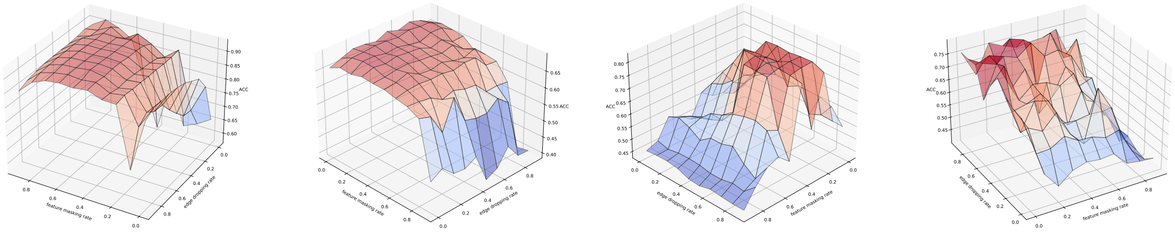
In this section, we analyse different components of MLG-CPC and results are presented in Table 4. As we can observe, the MLG-CPC w/o feature-level leads to severe degradation. Taking results on AMAP for example, our MLG-CPC exceeds the MLG-CPC w/o feature-level loss by 5.65%, 4.75%, 11.46%, 4.42% performance increment in terms of ACC, NMI, ARI and F1, which demonstrates the low-level yet fundamental feature representations are crucial for the high-level spaces. The MLG-CPC w/o prototypical-level loss also is inferior to the MLG-CPC, showing that the global prototypical information can benefit the model training. Besides, since the cluster loss plays essential role in our end-to-end framework, when we remove this sub-module, our MLG-CPC sharply degrades a lot, specifically, by 11.89%, 10.72%, 17.46%, 8.22% performance decrement with respect to ACC, NMI, ARI and F1, due to poor labels generated for clustering.

Dataset	Model	ACC	NMI	ARI	F1
ACM	w/o $\mathcal{L}_{fl}$	70.02 ± 3.9	32.36 ± 2.2	34.64 ± 2.5	69.95 ± 3.3
	w/o $\mathcal{L}_{pl}$	90.53 ± 1.1	70.03 ± 1.7	75.10 ± 1.6	90.50 ± 0.9
	w/o $\mathcal{L}_{cl}$	73.75 ± 4.1	34.47 ± 5.2	37.97 ± 4.7	73.89 ± 3.9
	fully model	**93.20 ± 0.1**	**75.57 ± 0.1**	**81.11 ± 0.1**	**93.16 ± 0.1**
DBLP	w/o $\mathcal{L}_{fl}$	76.78 ± 2.2	45.08 ± 1.5	49.67 ± 1.7	75.79 ± 2.0
	w/o $\mathcal{L}_{pl}$	80.93 ± 1.3	50.67 ± 2.5	57.15 ± 2.2	78.83 ± 1.7
	w/o $\mathcal{L}_{cl}$	59.30 ± 4.7	24.18 ± 4.3	23.91 ± 4.3	58.59 ± 4.8
	fully model	**82.13 ± 0.5**	**52.41 ± 0.6**	**57.78 ± 0.8**	**80.32 ± 0.5**
CITE	w/o $\mathcal{L}_{fl}$	59.62 ± 3.1	36.82 ± 2.2	35.52 ± 1.8	55.81 ± 2.9
	w/o $\mathcal{L}_{pl}$	65.28 ± 2.3	40.23 ± 1.7	41.66 ± 1.6	61.29 ± 1.9
	w/o $\mathcal{L}_{cl}$	47.55 ± 4.4	27.59 ± 3.6	29.96 ± 2.9	43.59 ± 3.8
	fully model	**69.31 ± 1.5**	**44.47 ± 0.6**	**45.66 ± 0.7**	**64.21 ± 1.3**
AMAP	w/o $\mathcal{L}_{fl}$	74.00 ± 2.9	63.82 ± 2.2	52.67 ± 2.0	71.10 ± 3.1
	w/o $\mathcal{L}_{pl}$	76.12 ± 1.7	65.58 ± 1.2	58.83 ± 1.3	73.32 ± 1.5
	w/o $\mathcal{L}_{cl}$	67.76 ± 3.3	57.85 ± 2.5	46.67 ± 2.7	67.30 ± 4.2
	fully model	**79.65 ± 0.9**	**68.57 ± 1.3**	**64.13 ± 1.2**	**75.52 ± 0.8**

Table 4: Ablation comparisons of MLG-CPC on four datasets.

4.5 Parameters Analysis

Graph augmentations play vital role in multi-view graph learning. To this end, we investigate the influence of different augmentation strategies for our MLG-CPC framework in



ACM CITE DBLP AMAP

Figure 3: Graph augmentations with different edge dropping and feature masking rates on four datasets, respectively.

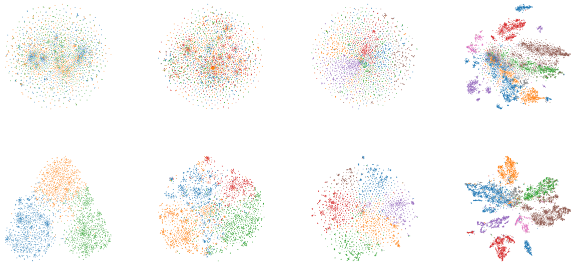
Figure 3, i.e., the edge dropping and feature masking with different probabilities, ranging from 0 to 1. We present the ACC results correspond to ACM, DBLP, CITE, and AMAP datasets. Since NMI, ARI, and F1-score have similar trends with ACC, we only present ACC due to the space limitations.

We can see our MLG-CPC is stable for most parameters settings, though there exists several severe performance degradation, we conjecture this due to two main causes. On the one hand, when structural and feature corruptions are larger than a relatively large value such as 0.8, the augmented (distorted) graphs have been overwhelmed in the entire framework, and thus encoders and sub-modules cannot be trained steadily without enough graph information, capturing limited and useless representations. On the other hand, we observe that poor capability happens when augmentations are slightly weak. For instance, when edge dropping rates are 0.1, performances degrade sharply on datasets. The reason behind this is augmented graphs across views in these cases change slightly. Since multi-view contrastive learning needs proper views to perform discrimination, this cannot provide sufficient information for multi-view learning. Overall, with moderate views, MLG-CPC can yield competitive performances, which demonstrates more elegant graph augmentations are significant for multi-view contrastive learning.

4.6 Visualization

We utilize t-SNE [Van der Maaten and Hinton, 2008] to visualize embedding of our MLG-CPC at different stages intuitively in Figure 4. At the initial stage, raw features are non-discriminative for clustering. During the model training phase, embedding gradually becomes more discriminative because objectives of MLG-CPC are optimized as epoch increases. At last, we can see the learned embedding is more promising, which has smaller intra-class distances as well as larger inter-class distances compared with raw features.

We plot similarity matrices of embedding on the ACM dataset in Figure 5. Compared with the matrix without feature orthogonality on the left, the embedding of MLG-CPC on the right is more approximate to the diagonal matrix. Thanks to this feature orthogonality, different dimensions are encouraged to capture distinct information, and thus the overlap information of different feature dimensions is minimized. This discriminative information will serve for the prototypical- and cluster-level space in the overall model, further enhancing the underlying performances of other sub-modules.



ACM DBLP CITE AMAP

Figure 4: 2D visualization on datasets. The first row and second row correspond to raw features and learned representations, respectively.

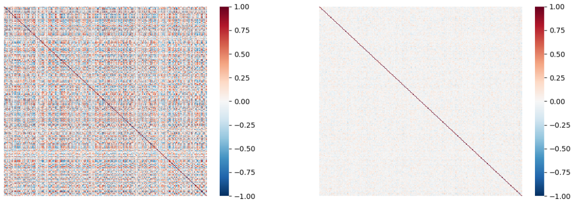


Figure 5: Similarity matrix of MLG-CPC without and with feature orthogonality on ACM dataset, respectively.

5 Conclusion

We introduce a novel graph contrastive framework for end-to-end clustering. Specifically, our MLG-CPC distills the features by orthogonality and generates assignments via cluster predictors. Currently, assignments are leveraged to generate prototypes, which then can perform prototype discrimination for semantic exploitation globally. Moreover, by virtue of different-level representation spaces, MLG-CPC concentrates on their objectives individually and jointly optimizes without laborious parameters balancing. Extensive experimental results on four datasets demonstrate the superiority of MLG-CPC. We will investigate the validity of MLG-CPC for graphs with heterophily in our future work.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant U21B2041, 61825603, National Key R&D Program of China 2020YFB2103902.

References

[Bo *et al.*, 2020] Deyu Bo, Xiao Wang, Chuan Shi, Meiqi Zhu, Emiao Lu, and Peng Cui. Structural deep clustering network. In *Proceedings of The Web Conference*, pages 1400–1410, 2020.

[Caron *et al.*, 2018] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision*, pages 132–149, 2018.

[Caron *et al.*, 2020] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924, 2020.

[Chen *et al.*, 2020] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pages 1597–1607, 2020.

[Ermolov *et al.*, 2021] Aleksandr Ermolov, Aliaksandr Siarohin, Enver Sangineto, and Nicu Sebe. Whitening for self-supervised representation learning. In *International Conference on Machine Learning*, pages 3015–3024, 2021.

[Gasteiger *et al.*, 2018] Johannes Gasteiger, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. In *International Conference on Learning Representations*, 2018.

[Gilmer *et al.*, 2017] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International Conference on Machine Learning*, pages 1263–1272, 2017.

[Gong *et al.*, 2022] Lei Gong, Sihang Zhou, Wenxuan Tu, and Xinwang Liu. Attributed graph clustering with dual redundancy reduction. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 3015–3021, 2022.

[Guo *et al.*, 2017] Xifeng Guo, Long Gao, Xinwang Liu, and Jianping Yin. Improved deep embedded clustering with local structure preservation. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1753–1759, 2017.

[Hamilton *et al.*, 2017] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in Neural Information Processing Systems*, 30:1025–1035, 2017.

[Hartigan and Wong, 1979] John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)*, 28(1):100–108, 1979.

[Hassani and Khasahmadi, 2020] Kaveh Hassani and Amir Hosein Khasahmadi. Contrastive multi-view representation learning on graphs. In *International Conference on Machine Learning*, pages 4116–4126, 2020.

[Keriven, 2022] Nicolas Keriven. Not too little, not too much: a theoretical analysis of graph (over) smoothing. *arXiv preprint arXiv:2205.12156*, 2022.

[Kipf and Welling, 2016] Thomas N Kipf and Max Welling. Variational graph auto-encoders. *Advances in Neural Information Processing Systems*, 2016.

[Kipf and Welling, 2017] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

[Li *et al.*, 2020] Xuelong Li, Han Zhang, Rong Wang, and Feiping Nie. Multiview clustering: A scalable and parameter-free bipartite graph fusion method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):330–344, 2020.

[Li *et al.*, 2021] Junnan Li, Pan Zhou, Caiming Xiong, and Steven CH Hoi. Prototypical contrastive learning of unsupervised representations. In *International Conference on Learning Representations*, 2021.

[Lin *et al.*, 2022] Shuai Lin, Chen Liu, Pan Zhou, Zi-Yuan Hu, Shuojia Wang, Ruihui Zhao, Yefeng Zheng, Liang Lin, Eric Xing, and Xiaodan Liang. Prototypical graph contrastive learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[Liu *et al.*, 2022] Yanfeng Liu, Qiang Li, Yuan Yuan, Qian Du, and Qi Wang. Abnet: Adaptive balanced network for multiscale object detection in remote sensing imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–14, 2022.

[Liu *et al.*, 2023] Yanfeng Liu, Zhitong Xiong, Yuan Yuan, and Qi Wang. Distilling knowledge from super resolution for efficient remote sensing salient object detection. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–16, 2023.

[Pan *et al.*, 2018] Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, Lina Yao, and Chengqi Zhang. Adversarially regularized graph autoencoder for graph embedding. In *International Joint Conference on Artificial Intelligence*, pages 2609–2615, 2018.

[Park *et al.*, 2019] Jiwoong Park, Minsik Lee, Hyung Jin Chang, Kyuewang Lee, and Jin Young Choi. Symmetric graph convolutional autoencoder for unsupervised graph representation learning. In *Proceedings of the International Conference on Computer Vision*, pages 6519–6528, 2019.

[Shchur *et al.*, 2018] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan

Günnemann. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868*, 2018.

[Snell *et al.*, 2017] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4080–4090, 2017.

[Topping *et al.*, 2022] Jake Topping, Francesco Di Giovanni, Benjamin Paul Chamberlain, Xiaowen Dong, and Michael M Bronstein. Understanding over-squashing and bottlenecks on graphs via curvature. In *International Conference on Learning Representations*, 2022.

[Tu *et al.*, 2021] Wenxuan Tu, Sihang Zhou, Xinwang Liu, Xifeng Guo, Zhiping Cai, En Zhu, and Jieren Cheng. Deep fusion clustering network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 9978–9987, 2021.

[Van der Maaten and Hinton, 2008] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(11), 2008.

[Veličković *et al.*, 2018] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

[Velickovic *et al.*, 2019] Petar Velickovic, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. In *International Conference on Learning Representations*, 2019.

[Wang and Isola, 2020] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pages 9929–9939, 2020.

[Wang *et al.*, 2019] Chun Wang, Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, and Chengqi Zhang. Attributed graph clustering: a deep attentional embedding approach. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 3670–3676, 2019.

[Wang *et al.*, 2022a] Qi Wang, Yanfeng Liu, Zhitong Xiong, and Yuan Yuan. Hybrid feature aligned network for salient object detection in optical remote sensing imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–15, 2022.

[Wang *et al.*, 2022b] Qi Wang, Yanling Miao, Mulin Chen, and Yuan Yuan. Spatial-spectral clustering with anchor graph for hyperspectral image. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–13, 2022.

[Xia *et al.*, 2022] Jun Xia, Lirong Wu, Ge Wang, Jintao Chen, and Stan Z Li. Progcl: Rethinking hard negative mining in graph contrastive learning. In *International Conference on Machine Learning*, pages 24332–24346, 2022.

[Xie *et al.*, 2016] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International Conference on Machine Learning*, pages 478–487, 2016.

[Xu *et al.*, 2021] Minghao Xu, Hang Wang, Bingbing Ni, Hongyu Guo, and Jian Tang. Self-supervised graph-level representation learning with local and global structure. In *International Conference on Machine Learning*, pages 11548–11558, 2021.

[Yang *et al.*, 2017] Bo Yang, Xiao Fu, Nicholas D Sidiropoulos, and Mingyi Hong. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *International Conference on Machine Learning*, pages 3861–3870, 2017.

[Yuan *et al.*, 2018] Yuan Yuan, Jie Fang, Xiaoqiang Lu, and Yachuang Feng. Remote sensing image scene classification using rearranged local features. *IEEE Transactions on Geoscience and Remote Sensing*, 57(3):1779–1792, 2018.

[Zbontar *et al.*, 2021] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320, 2021.

[Zhu *et al.*, 2021] Hao Zhu, Ke Sun, and Peter Koniusz. Contrastive laplacian eigenmaps. *Advances in Neural Information Processing Systems*, 34:5682–5695, 2021.