Dual Contrastive Graph-Level Clustering with Multiple Cluster Perspectives Alignment

Jinyu Cai[1], **Yunhe Zhang**[2,3], **Jicong Fan**[2,3], **Yali Du**[4] and **Wenzhong Guo**[5*]
[1]Institute of Data Science, National University of Singapore, Singapore
[2]Shenzhen Research Institute of Big Data, Shenzhen, China
[3]School of Data Science, The Chinese University of Hong Kong, Shenzhen (CUHK-Shenzhen), China
[4]Department of Informatics, King's College London, United Kingdom
[5]College of Computer and Data Science, Fuzhou University, China
jinyucai@nus.edu.sg, zhangyhannie@gmail.com, fanjicong@cuhk.edu.cn, yali.du@kcl.ac.uk, guowenzhong@fzu.edu.cn

Abstract

Graph-level clustering, which is essential in medical, biomedical, and social network data analysis, aims to group a set of graphs into various clusters. However, existing methods generally rely on a single clustering criterion, e.g., k-means, which limits their abilities to fully exploit the complex Euclidean and structural information inherent in graphs. To bridge this gap, we propose a dual contrastive graph-level clustering (DCGLC) method in this paper. DCGLC leverages graph contrastive learning and introduces the Euclidian-based and subspace-based cluster heads to capture the cluster information from different cluster perspectives. To overcome the inconsistency estimations and fuse the cluster information of multiple cluster heads, we propose a contrastive mechanism to align the cluster information derived from them. The cluster-perspective contrast facilitates the capture of more comprehensive cluster information. Importantly, DCGLC is an end-to-end framework in which graph contrastive learning and cluster-perspective contrast are mutually improved. We demonstrate the superiority of DCGLC over the state-of-the-art baselines on numerous graph benchmarks.

1 Introduction

Graph-structured data is one of the most common data structures in the real world, and some examples include molecular data [Rong *et al.*, 2020], biological data [Agarwal, 2006], and social networks [Aggarwal, 2011], etc. Clustering on graph data [Schaeffer, 2007; Li *et al.*, 2023] is a fundamental research topic in machine learning, which aims to partition the nodes of a single graph (node-level) or a set of graphs (graph-level) into different groups without supervision information. Early works on graph data clustering were mainly based on graph cut [Schaeffer, 2007] and spectral clustering (SC) [Ng *et al.*, 2001; Fan *et al.*, 2022; Chen *et al.*, 2022; Chen *et al.*, 2023a], but such approaches are difficult and time-consuming to handle complex graph structures or very large graphs. Therefore, learning representative features for graph data is an intuitive way to improve the efficiency and performance of graph clustering.

The upsurge of deep learning and graph neural networks (GNNs) [Wu *et al.*, 2020] facilitate the progress of graph clustering. Many GNN-based deep clustering approaches [Wang *et al.*, 2017; Bo *et al.*, 2020; Tu *et al.*, 2021; Liu *et al.*, 2022] have exhibited remarkable success in node clustering. Recently, graph contrastive learning (GCL) [Zhu *et al.*, 2021a] has attracted significant interest from researchers due to its powerful feature learning capability. GCL learns the representation of graph data by imposing various forms of augmentation to the graphs and then contrasting between positive and negative pairs. By introducing the concept of GCL into learning node features, several GCL-based graph clustering methods [Zhao *et al.*, 2021; Xia *et al.*, 2022; Liu *et al.*, 2023b; Liu *et al.*, 2023a] have further raised the performance bar of node clustering.

However, there are still very limited efforts delving into graph-level clustering. This disparity stems from the fact that node-level clustering operates within a single graph, where node representations can be learned more easily through information aggregation between nodes and their neighbors [Wu *et al.*, 2020]. Conversely, graph-level tasks are specific to a series of graphs, which presents a more complex and challenging task of measuring the relationships between them. Recent advances in graph-level tasks have focused on graph-level classification, mostly based on graph kernels and graph-level representation learning methods [Yang *et al.*, 2023]. Graph kernels [Vishwanathan *et al.*, 2010; Sun and Fan, 2024], such as the Weisfeiler-Lehman kernel (WL) [Shervashidze *et al.*, 2011] and Short-Path kernel (SP) [Borgwardt and Kriegel, 2005], measure similarity between graphs using well-defined kernel functions. While graph representation learning [Xu *et al.*, 2019; Sun *et al.*, 2020; You *et al.*, 2020; Cai *et al.*, 2024a; Wu *et al.*, 2024b] generally leverages GNNs [Kipf and Welling, 2017; Xu *et al.*, 2019; Wu *et al.*, 2024a] as the backbone, together with readout functions, to aggregate node features for learning graph-

*Corresponding author.

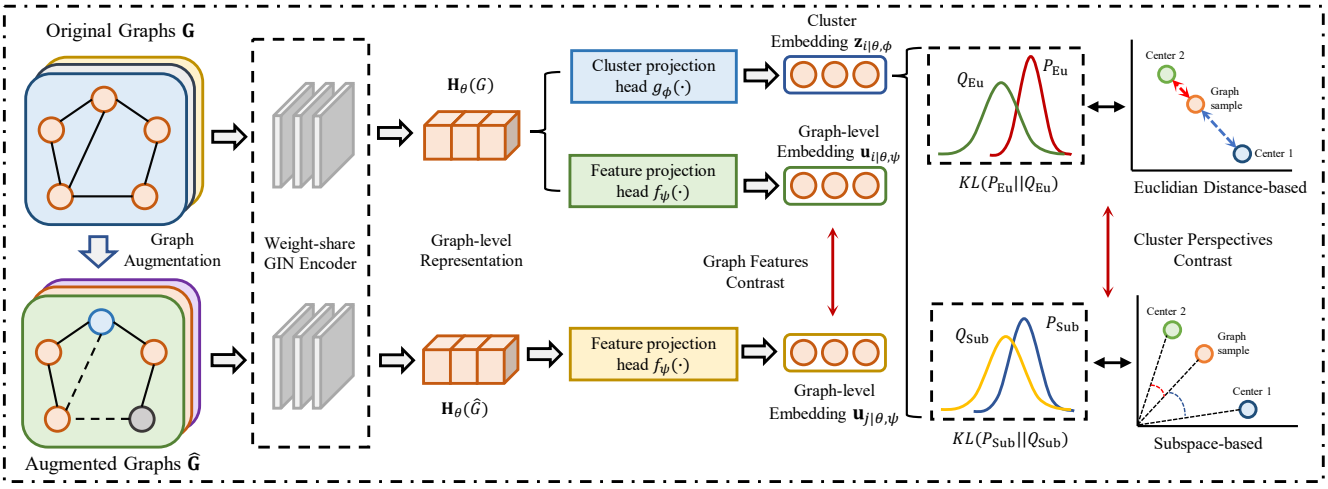


Figure 1: An illustration of the proposed method. Note that two arrows marked in **red** denote the two main objectives in our method, i.e., graph-level features contrast and cluster-perspective contrast.

level features. Graph-level classification is then achieved via a two-stage strategy that combines a classifier with either graph kernels or graph representation learning methods.

Nevertheless, these approaches tend to be a trivial solution when implemented in graph-level clustering, primarily due to the following challenges.

- supervision information is used to train the classifier, whereas clustering is not. Consequently, graph-level clustering demands a much higher requirement for the discriminability of learned features than graph-level classification.
- The aforementioned approaches typically construct as a two-stage model, where the learned similarity matrix or graph-level representations cannot be optimized for specific clustering tasks. Previous works on deep clustering [Yang *et al.*, 2016; Xie *et al.*, 2016; Cai *et al.*, 2022a] have demonstrated the benefits of incorporating an explicit clustering objective into feature learning.
- Relying on a single cluster perspective hardly exploits comprehensive cluster information. Different cluster perspectives may reveal different cluster information. For example, k-means [Hartigan and Wong, 1979] focuses more on the Euclidean distance relationship between data, while SC [Ng *et al.*, 2001] highlights connectivity and reveals clusters under subspace structures.

These challenges naturally pose an intriguing research question: *Can we advance graph-level clustering by designing a joint optimization framework and leveraging information from different cluster perspectives?*

To answer this question, we propose a dual contrastive graph-level clustering (DCGLC)[1] in this paper. Figure 1 shows the network structure of DCGLC. Specifically, we leverage graph contrastive learning to enhance the effectiveness of graph-level representations by maximizing the consistency between positive sample pairs and inconsistency between negative sample pairs. Additionally, we introduce an explicit clustering objective into graph contrastive learning, which enables our model to be optimized for clustering and learns discriminative graph-level representations. Furthermore, we propose a contrastive mechanism to align clustering information from both the Euclidean-based and subspace-based cluster perspectives, where the former focuses on the geometric relationships between graphs while the latter captures their structural relationships. The proposed DCGLC encourages the incorporation of clustering information from different cluster perspectives and ensures the consistency of clustering assignment across them, which leads to more reliable clustering results. Besides, the graph-level features contrast, and cluster-perspective contrast can be mutually improved, facilitating the learning of cluster-friendly representations. We summarize the main contributions of this work as follows:

- We propose DCGLC, an end-to-end clustering method unifying the optimization of graph contrastive learning with information integration of different cluster perspectives.
- We propose a cluster-perspective contrast mechanism to align the cluster information, which results in more reliable clustering assignments by considering both geometric and structural relationships between graphs.
- We demonstrate the effectiveness of DCGLC through extensive experiments on various types of graph benchmarks. Experimental results show that DCGLC outperforms state-of-the-art baselines in graph-level clustering.

2 Related Work

Deep Graph Clustering. Clustering [Xie *et al.*, 2016; Zhang *et al.*, 2022; Cai *et al.*, 2022b; Fu *et al.*, 2022; Yao *et al.*, 2023; Zhang *et al.*, 2023; Cai *et al.*, 2024b] is a fundamental task in machine learning, and is crucial for real-world data analysis. With the upsurge of GNNs, deep graph clustering, especially node clustering, has received increasing attention and advancement in recent years. For instance,

[1]Code is available at: https://github.com/wownice333/DCGLC

Wang *et al.* [2017] proposed a marginalized graph autoencoder, which leverages both structural information and intrinsic properties of graph data to learn effective node features for clustering. Bo *et al.* [2020] combined graph convolutional network (GCN) [Kipf and Welling, 2017; Chen *et al.*, 2023b; Chen *et al.*, 2023c] and an integrated clustering module to learn effective node features for graph clustering. Tu *et al.* [2021] proposed a dynamic fusion mechanism to fuse the structural information and attributes and further designed a triplet self-supervised strategy to produce reliable cluster assignment. As an emerging technique, GCL-based methods [Zhao *et al.*, 2021; Pan and Kang, 2021; Liu *et al.*, 2022; Liu *et al.*, 2023b] are also widely studied. Zhao *et al.* [2021] proposed a contrastive node clustering framework that encourages the similarity of positive pairs generated by attribute masks, and the dissimilarity of negative pairs constituted from out-of-cluster samples. Pan and Kang [2021] leveraged contrastive learning to learn consensus representation from multiple views for node clustering. Liu *et al.* [2023b] proposed a contrastive graph clustering method that effectively mines hard samples in clustering. Despite remarkable advancements in node clustering, a substantial research gap persists in graph-level clustering, because measuring the relationships between graphs is more complicated than nodes within a single graph.

Graph Representation Learning. Graph representation learning [Xu *et al.*, 2019; Wu *et al.*, 2020; Sun *et al.*, 2024] aims to learn graph-level representation by aggregating node features learned with powerful GNNs, and has witnessed remarkable progress. For example, Sun *et al.* [2020] maximized the mutual information between the global and substructural information of data to learn comprehensive graph-level representations. You *et al.* [2020] proposed GraphCL, a graph augmentation mechanism to learn effective graph-level representations based on GCL [Zhu *et al.*, 2021a] for graph classification. We also follow the graph augmentation strategies proposed in [You *et al.*, 2020] in this study, e.g., node dropping, edge perturbation, attribute masking, and subgraph, to augment graph data. You *et al.* [2021] proposed JOAO to investigate the automatic graph augmentation mechanisms in GCL. However, the majority of existing approaches focused on classification tasks and still very limited efforts on graph-level clustering. Besides, these approaches usually construct two-stage models, which poses a greater challenge for graph-level clustering as no supervision information can be utilized.

3 Proposed Method

3.1 Problem Formulation

Let $\mathbf{G} = \{G_1, \ldots, G_N\}$ be a set of N graphs and each graph is formulated as $G_i = \{V_i, E_i, \{\mathbf{x}_v^{(i)}\}_{v \in V_i}\}$, where V_i is the node set, E_i is the edge set, and $\mathbf{x}_v^{(i)}$ is the feature vector of node v on graph i. We regard the problem that aims to partition the graph set into C different groups without any supervision information as *graph-level clustering*, i.e., to let $\mathbf{G} = \mathbf{G}^{(1)} \cup \mathbf{G}^{(2)} \cdots \cup \mathbf{G}^{(C)}, \text{s.t. } \mathbf{G}^{(i)} \cap \mathbf{G}^{(j)} = \varnothing, \forall i \neq j$. Different from the graph-level classification and node-level clustering [Wu *et al.*, 2020], graph-level clustering is still a challenging task in machine learning and has rarely been studied [Xu *et al.*, 2022] because measuring the relationships between graphs in an unsupervised setting is much more complex than: 1) nodes within a single graph and 2) in a supervised setting. Therefore, the important issue in graph-level clustering is the exploration of learning graph-level features that contain clustering information and are sufficiently discriminative. To this end, we propose to achieve graph-level clustering by solving the following general problem:

$$\underset{\theta,\psi,\phi,\mathbf{G}}{\text{minimize}} \mathop{\mathbb{E}}_{G \sim \mathbf{G}} \ell_{r|\psi}(h_\theta(G)) + \mathop{\mathbb{E}}_{G \sim \mathbf{G}} \ell_{c|\phi}(h_\theta(G)), \tag{1}$$

where $\ell_{r|\psi}(\cdot)$ denotes the representation learning objective with the aim of learning effective graph-level features, and $\ell_{c|\phi}(\cdot)$ is the clustering objective with the aim of capturing cluster information. A GNN [Kipf and Welling, 2017; Zhou *et al.*, 2020] based model $h_\theta(\cdot)$, which has become a paradigm for many graph-based tasks [Xu *et al.*, 2019; Park and Neville, 2019; Bo *et al.*, 2020], is generally leveraged to facilitate the learning of graph representations from graph data. In the following section, we will show how to solve (1) and achieve graph-level clustering through the joint training of learning effective graph-level representation and leveraging comprehensive cluster information.

3.2 Graph-Level Representation Contrast

We first present the representation learning module, where we attempt to explore the capability of GCL [Zhu *et al.*, 2021a] in learning effective representation for graph-level clustering. As an emerging technique in graph representation learning, GCL-based methods surpass state-of-the-art mutual information (MI) maximization-based methods [Veličković *et al.*, 2019; Sun *et al.*, 2020] in many tasks such as graph-level classification [You *et al.*, 2020; Zhu *et al.*, 2021b] and node-level clustering [Zhao *et al.*, 2021; Xia *et al.*, 2022]. Specifically, given a graph G_j, the learned features of node i in the k-th GNN layer can be defined as:

$$\mathbf{h}_i^{(k)} = \text{COMBINE}^{(k)}(\mathbf{h}_i^{(k-1)}, \mathbf{a}_i^{(k)}), \tag{2}$$

where $\mathbf{a}_i^{(k)}$ denotes the aggregated neighbour features of node i, i.e., $\mathbf{a}_i^{(k)} = \text{AGGREGATE}^{(k)}(\mathbf{h}_v^{(k-1)} : v \in \mathcal{N}(i))$, and $\mathcal{N}(i)$ is the neighbor set of node i. $\text{COMBINE}(\cdot)$ indicates the operation of integrating the information between node i and its neighbours. In particular, the initial feature $\mathbf{h}_i^{(0)} = \mathbf{x}_i$. Subsequently, the graph-level representation of graph G_j, i.e., $\mathbf{H}_\theta(G_j)$, can be formulated as follows:

$$\mathbf{H}_\theta(G_j) = \text{READOUT}(\{\mathbf{h}_\theta^i\}_{i=1}^{n_{G_j}}), \tag{3}$$

where $\mathbf{h}_\theta^i$ denotes the concatenated features of node i among all K layers, i.e., $\mathbf{h}_\theta^i = \text{CONCAT}(\{\mathbf{h}_i^{(k)}\}_{k=1}^K)$, and n_{G_j} is the node numbers of graph G_j. Note that the $\text{READOUT}(\cdot)$ function (e.g., Sum, Average, or Max/Min) aims to integrate the local features within a graph to obtain graph-level features [Kipf and Welling, 2017; Xu *et al.*, 2019], and we utilize the sum readout function in our implementation.

Given a graph G, we follow the graph augmentation strategy $\mathcal{A}$ in [You *et al.*, 2020] to produce an augmented graph

$\hat{G}_i \sim \mathcal{A}_i(\hat{G}_i|G)$. Note that a minibatch with N_b graphs in training will generate N_b augmented graphs. Therefore, the original graph G_i and its corresponding augmented graph $\hat{G}_i$ can be regarded as a positive pair in contrastive learning, and the combination of G_i with other augmented graphs $\{\hat{G}_j, i \neq j\}$ can be regarded as negative pairs then. For each graph pair such as G_i and $\hat{G}_i$, their graph-level representations $\mathbf{H}_\theta(G_i)$, $\mathbf{H}_\theta(\hat{G}_i)$ can be obtained through a GNN encoder following Eqs. (2) and (3). Besides, the contrastive loss is generally calculated in the low-dimensional latent space, therefore a projection head $f_\psi(\cdot)$ is applied to project the graph-level representation $\mathbf{H}_\theta(G_i)$, $\mathbf{H}_\theta(\hat{G}_i)$ into $\mathbf{u}_{i|\theta,\psi}$ and $\hat{\mathbf{u}}_{i|\theta,\psi}$. We use NT-Xent loss [Sohn, 2016] as the contrastive loss in this paper, the NT-Xent loss is to maximize the consistency among positive pairs and the inconsistency among negative pairs. It is worth noting that we do not further generate other negative samples, instead, we regard the other $N_b - 1$ augmented samples in a minibatch as negative samples. For the N_b graphs in a minibatch, we aim to align the learned graph-level representations from the two sets of augmented graphs, which can be achieved by minimizing

$$\ell_{\mathrm{r}|\theta,\psi} = -\frac{1}{N_b}\sum_{i=1}^{N_b}\log\frac{\exp(\mathrm{sim}(\mathbf{u}_{i|\theta,\psi}, \hat{\mathbf{u}}_{j|\theta,\psi})/\tau)}{\sum_{j=1,j\neq i}^{N_b}\exp(\mathrm{sim}(\mathbf{u}_{i|\theta,\psi}, \hat{\mathbf{u}}_{j|\theta,\psi})/\tau)} \quad (4)$$

where $\mathrm{sim}(\mathbf{u}_{i|\theta,\psi}, \hat{\mathbf{u}}_{j|\theta,\psi}) = \frac{\mathbf{u}_{i|\theta,\psi}^\top \hat{\mathbf{u}}_{j|\theta,\psi}}{\|\mathbf{u}_{i|\theta,\psi}\|\|\hat{\mathbf{u}}_{j|\theta,\psi}\|}$ denotes the cosine similarity function, and τ indicates the temperature parameter. The loss function aims to minimize the distance between feature vectors of the same samples while maximizing the distance between feature vectors of different samples, resulting in learning more discriminative graph-level representations. Apparently, an intuitive way to achieve graph-level clustering is to directly perform k-means [Hartigan and Wong, 1979] or SC [Ng *et al.*, 2001] in the learned graph-level representation. However, there are two important issues in this solution:

1. Graph-level representation learning and clustering are two separate processes without any explicitly defined clustering objective, which implies that they may fail to learn cluster-friendly graph-level representations. The empirical results in the experiment section also demonstrate that those two-stage clustering models, i.e., the simple combination of k-means/SC and graph kernels/graph representation learning, tend to be trivial solutions.
2. The Euclidean-based (e.g., k-means) and subspace-based clustering (e.g., SC) criteria offer cluster information from different perspectives. Euclidean-based methods prioritize distance information between data, while subspace-based methods emphasize the angle information. Combining these two perspectives can potentially provide richer information for graph-level clustering.

3.3 Multiple Cluster Perspectives Contrast

To address these issues, we propose a cluster-perspective contrast approach that aligns clustering information derived from multiple perspectives, which also introduces an explicit clustering objective to guide the optimization. Specifically, inspired by [Xie *et al.*, 2016; Cai *et al.*, 2022a], we cluster from two perspectives by utilizing two clustering heads, i.e., 1) Euclidian-based and 2) subspace-based, respectively. It should be noted that the clustering task generally performs in a low-dimensional embedding space, therefore we further project the graph-level features $\mathbf{H}_\theta(G_i)$ into the cluster embedding $\mathbf{Z}_{\theta,\phi}$ with a projector $g_\phi(\cdot)$ parameterized by ϕ. The Euclidian-based clustering head measures the discrepancy between the graph-level features and each cluster center with Euclidian distance, which can be defined as follows:

$$q^{\mathrm{Eu}}_{ic|\theta,\phi} = \frac{(1+\|\mathbf{z}_{i|\theta,\phi} - \boldsymbol{\mu}_c\|^2)^{-1}}{\sum_{c=1}^{C}(1+\|\mathbf{z}_{i|\theta,\phi} - \boldsymbol{\mu}_c\|^2)^{-1}}, \quad (5)$$

where $\boldsymbol{\mu}_c$ is the c-th cluster center of C clusters in total and can be initialized by k-means. $\mathbf{z}_{i|\theta,\phi} = g_\phi(\mathbf{H}_\theta(G_i)) \in \mathbb{R}^{d_z}$ is the cluster embedding for graph G_i and d_z is the dimension of cluster embedding. Another clustering head is based on subspace clustering, with the assumption that the data can be divided by several subspaces when they are projected into a low-dimensional embedding space. Consequently, the subspace-based clustering head can be formulated by the similarity between the cluster embedding and the defined subspace base as follows:

$$q^{\mathrm{Sub}}_{ic|\theta,\phi} = \frac{\|\mathbf{z}_{i|\theta,\phi}^\top \mathbf{D}^{(c)}\|_F^2 + \eta d}{\sum_{c=1}^{C}(\|\mathbf{z}_{i|\theta,\phi}^\top \mathbf{D}^{(c)}\|_F^2 + \eta d)}, \quad (6)$$

where $\mathbf{D} \in \mathbb{R}^{Cd\times d}$ is the subspace base with d-dimension. It can be initialized by the column space of clusters generated by performing k-means on $\mathbf{Z}_{\theta,\phi}$. $\|\mathbf{z}_{i|\theta,\phi}^\top \mathbf{D}^{(c)}\|_F^2$ quantifies the 'closeness' (cosine similarity) between the graph-level feature $\mathbf{z}_{i|\theta,\phi}$ and the c-th subspace $\mathbf{D}^{(c)}$. η is the smooth parameter and is fixed as 2. Particularly, the subspace $\mathbf{D}$ should obey two rules: 1) $\|\mathbf{D}_i^{(c)}\| = 1, i = 1,\ldots,d, c = 1,\ldots,C$; 2) $\|\mathbf{D}^{(i)\top}\mathbf{D}^{(j)}\|_F \leq \tau, i \neq j, \tau \to 0$. In practice, we impose two constraints by:

$$\ell_{\mathrm{cons}} = \frac{1}{2}(\|\mathbf{D}^\top\mathbf{D}\odot\mathbf{I} - \mathbf{I}\|_F^2 + \|\mathbf{D}^\top\mathbf{D}\odot\mathbf{O}\|_F^2), \quad (7)$$

where $\odot$ denotes the Hadamard product, $\mathbf{I}$ is an identity matrix with the same size as $\mathbf{D}$. $\mathbf{O}$ is a special matrix whose elements in each d-size block are 0 and the others are 1. The first term ensures each base vector is of unit length, preventing the value of $\max\|\mathbf{z}_{j|\theta,\phi}^\top \mathbf{D}^{(c)}\|_F^2$ from being zero. The second term encourages the subspace bases to be mutually orthogonal, which is critical for maintaining distinct and separate subspaces.

After defining the learnable cluster assignment Q_{Eu} and Q_{Sub} from the distance and subspace perspectives, we can further yield a refined assignment, e.g., P_{Eu}, defined as:

$$p^{\mathrm{Eu}}_{ic} = \frac{(q^{\mathrm{Eu}}_{ic|\theta,\phi})^2/\sum_{i=1}^{N_b} q^{\mathrm{Eu}}_{ic|\theta,\phi}}{\sum_{c=1}^{C}((q^{\mathrm{Eu}}_{ic|\theta,\phi})^2/\sum_{i=1}^{N_b} q^{\mathrm{Eu}}_{ic|\theta,\phi})}, \quad (8)$$

where N_b represents the number of graphs in a batch, and the effect of P_{Eu} is to emphasize the high-confidence assignments in Q_{Eu}. Similarly, we can derive P_{Sub} from Q_{Sub} via:

$$p^{\mathrm{Sub}}_{ic} = \frac{(q^{\mathrm{Sub}}_{ic|\theta,\phi})^2/\sum_{i=1}^{N_b} q^{\mathrm{Sub}}_{ic|\theta,\phi}}{\sum_{c=1}^{C}((q^{\mathrm{Sub}}_{ic|\theta,\phi})^2/\sum_{i=1}^{N_b} q^{\mathrm{Sub}}_{ic|\theta,\phi})}. \quad (9)$$

Algorithm 1 Training flows of DCGLC.

Input: Graph set $\mathbf{G}$, number of clusters C, batch size N_b, learning rate α, total training epochs $\mathcal{T}$.
Output: The clustering results $\mathcal{C}$.

```
Initialize the parameters θ, φ, ψ and cluster centers μ;
while not converge do
    for t = 1, 2, ..., T do
        Generate augmented graph set Ĝ from G;
        Obtain graph representations H_θ(G_i) and H_θ(Ĝ_i) with Eq. (2) and (3);
        Obtain the cluster embeddings Z_i, and cluster assignments Q_Eu and Q_Sub with Eqs. (5) and (6);
        Compute refined cluster assignments P_Eu and P_Sub via Eqs. (8) and (9);
    end for
    Compute the total loss with Eqs. (4), (7), (10), (11), and (12);
    Back-propagation and update network parameters θ, φ, ψ, and the cluster centers μ;
end while
Compute final cluster assignment Q_Eu and Q_Sub;
Obtain the clustering labels with Eq. (13);
return The clustering results C.
```

Note that the refined assignments serve as the pseudo labels to provide guidance for the update of Q_{Eu} and Q_{Sub}, which can be optimized by minimizing the discrepancy between the cluster assignment distribution and the refined assignment distribution, i.e.:

$$\begin{aligned}\ell_{\text{KL}|\theta,\phi} &= \text{KL}(P_{\text{Eu}}||Q_{\text{Eu}}) + \text{KL}(P_{\text{Sub}}||Q_{\text{Sub}}) \\ &= \sum_{i=1}^{N_b}\sum_{c=1}^{C} p_{ic}^{\text{Eu}} \log \frac{p_{ic}^{\text{Eu}}}{q_{ic|\theta,\phi}^{\text{Eu}}} + \sum_{i=1}^{N_b}\sum_{c=1}^{C} p_{ic}^{\text{Sub}} \log \frac{p_{ic}^{\text{Sub}}}{q_{ic|\theta,\phi}^{\text{Sub}}},\end{aligned} \tag{10}$$

where $\text{KL}(\cdot||\cdot)$ denotes the Kullback-Leibler (KL) divergence between two distributions. As previously discussed, our objective is to learn reliable and consistent clustering assignments by incorporating more comprehensive cluster information from multiple cluster perspectives (i.e., Euclidian-based one denoted by Q_{Eu} and subspace-based one denoted by Q_{Sub}) for each graph. However, clustering from different perspectives may yield inconsistent results due to diverse evaluation criteria, which ultimately affects the consistency of clustering and leads to suboptimal clustering performance. To address this challenge, we propose a cluster-perspective contrast objective $\ell_{\text{c}|\theta,\phi}$ for the cluster information alignment of different perspectives, which is defined as follows:

$$\ell_{\text{c}|\theta,\phi} = -\frac{1}{C}\sum_{i=1}^{C} \log \frac{\exp(\text{sim}(q_{i|\theta,\phi}^{\text{Eu}}, q_{j|\theta,\phi}^{\text{Sub}})/\tau)}{\sum_{j=1,j\neq i}^{C} \exp(\text{sim}(q_{i|\theta,\phi}^{\text{Eu}}, q_{j|\theta,\phi}^{\text{Sub}})/\tau)}, \tag{11}$$

where C denotes the number of clusters, and the definitions of τ and $\text{sim}(\cdot,\cdot)$ are same to Eq. (4). Notably, we also demonstrate that aligning different cluster perspectives outperforms solely utilizing a single clustering head with a comprehensive experiment via an ablation study (in Section 4.2).

3.4 Training Strategy

The two modules in the proposed method are jointly optimized by minimizing the following total objective function:

$$L = \ell_{\text{r}|\theta,\psi} + \lambda\ell_{\text{c}|\theta,\phi} + \beta\ell_{\text{KL}|\theta,\phi} + \ell_{\text{cons}}, \tag{12}$$

where λ and β are trade-off parameters that control the contributions of $\ell_{c|\theta,\phi}$ and $\ell_{\text{KL}|\theta,\phi}$ to the total loss. Compared to (1), we not only comprise a well-defined representation learning module and clustering module, but also incorporate cluster information from different cluster perspectives. The dual contrastive learning in graph-level feature pairs and multiple cluster perspectives allows us to obtain more reliable cluster assignments. More importantly, our approach stands out as an end-to-end graph-level clustering method that is able to produce clustering results directly from the network, eliminating the reliance on performing other clustering methods (e.g., k-means and SC) on the learned graph-level representations. The clustering results $\mathcal{C}$ for each graph can be obtain from the trained Q_{Eu} and Q_{Sub} by:

$$\mathcal{C} = \arg\max(\max(Q_{\text{Eu}}, Q_{\text{Sub}})). \tag{13}$$

Algorithm 1 summarizes the detailed training flows of the proposed DCGLC.

3.5 Complexity Analysis

Here, we further theoretically analyze the time and space complexity for the proposed DCGLC method. Considering N graph with average n nodes and $|E|$ edges per graph, the number of GIN layers and clusters are K and C, and the dimensions of nodes and each hidden layer are m and d_h. We analyze the time and space complexity of DCGLC as follows:

- The time complexity of the first layer and the other layers in GIN are approximate $\mathcal{O}(Nn|E|md_h)$ and $\mathcal{O}(Nn|E|d_h^2)$, where $|E|$ typically can be ignored when the adjacency matrix is sparse. Besides, the time complexity of each cluster head is $\mathcal{O}(NKd_hC)$. Therefore, the time complexity of DCGLC is $\mathcal{O}(Nd_h(nm + (K-1)nd_h + 2KC))$.
- Regarding space complexity, the GIN backbone in DCGLC needs to store the neighbor list and feature matrices of the entire graph to represent the graph structure. Thus the space complexity of these processes is $\mathcal{O}(Nn + N|E| + Nnm)$. Besides, the computation process of the GIN needs to store the intermediate results of each GIN layer. Thus the space complexity of this process for the GIN with K layers is $\mathcal{O}(Nnd_h(m + (K-1)d_h))$. The space complexity of the two cluster heads is $\mathcal{O}(NKd_hC)$. Overall, the space complexity of DCGLC is $\mathcal{O}(Nn+N|E|+Nnm+Nnd_h(m+(K-1)d_h)+2NKd_hC)$.

From the above discussion, we can observe that the time and space complexities of DCGLC are mainly associated with the number of nodes and graphs, i.e., n and N, which is competitive to GCL-based baselines such as GraphCL [You *et al.*, 2020], JOAO [You *et al.*, 2021], and GLCC [Ju *et al.*, 2023].

3.6 Discussion with Previous Works

Numerous works have paved the way for this work, and we briefly discuss the connections between our work and them.

Method	MUTAG			BZR			DD			IMDB-BINARY		
	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
SP	72.87±0.00	10.24±0.00	15.95±0.00	79.51±0.00	4.13±0.00	3.97±0.00	58.83±0.00	0.43±0.00	0.14±0.00	53.95±1.15	6.52±1.69	0.66±0.36
GK	67.02±0.00	1.74±0.00	1.04±0.00	61.23±3.36	1.06±1.21	3.13±3.74	57.60±1.49	0.28±0.13	-0.05±0.55	50.20±0.00	0.39±0.00	0.00±0.00
RW	77.65±0.00	30.81±0.00	30.26±0.00	64.69±0.00	0.00±0.00	-0.15±0.00	OM	OM	OM	51.30±0.00	0.16±0.00	0.03±0.00
WL	73.40±0.00	14.50±0.00	21.20±0.00	75.56±0.00	0.50±0.00	3.76±0.00	58.57±0.00	0.13±0.00	-0.05±0.00	51.24±0.53	0.69±0.79	0.06±0.05
LT	56.60±4.88	3.09±1.38	-0.62±0.63	78.35±0.35	0.69±0.28	1.12±1.03	OM	OM	OM	51.20±0.00	1.61±0.00	0.05±0.00
WL-OA	67.55±0.00	19.64±0.00	11.40±0.00	69.63±0.00	5.60±0.00	-8.67±0.00	OM	OM	OM	50.86±0.52	0.53±0.84	0.02±0.05
InfoGraph+KM	77.95±1.41	35.22±3.47	30.95±3.03	63.62±2.41	1.59±0.95	2.39±1.44	58.57±0.04	0.64±0.00	-0.02±0.08	54.79±0.84	4.77±0.16	0.92±0.38
InfoGraph+SC	72.58±4.83	28.68±4.93	19.85±5.91	73.53±2.66	3.66±2.52	5.04±3.12	58.83±0.00	0.43±0.00	0.14±0.00	54.90±3.69	4.31±1.91	1.41±2.16
GraphCL+KM	77.07±1.21	35.69±2.83	28.99±2.65	71.43±4.09	1.04±0.77	3.07±1.03	58.02±0.04	0.67±0.00	-0.33±0.03	54.66±0.13	5.16±0.19	0.83±0.02
GraphCL+SC	73.22±2.66	32.19±2.05	23.44±2.45	72.88±1.66	1.90±0.38	3.47±0.59	55.18±2.21	0.19±0.17	0.62±0.67	57.42±1.35	5.01±1.34	2.76±1.56
JOAO+KM	79.20±0.72	36.32±3.03	33.74±1.65	72.64±4.26	1.37±1.14	4.01±3.39	57.95±0.04	1.49±0.00	-0.37±0.00	58.20±1.84	6.76±1.19	2.77±1.13
JOAO+SC	70.72±2.85	27.73±0.23	17.12±2.03	72.98±1.59	2.75±1.30	5.62±3.74	56.28±0.48	0.33±0.00	1.01±0.00	50.20±0.00	0.39±0.00	0.00±0.00
GWF+KM	66.94±7.68	12.46±9.31	13.32±10.53	53.00±0.31	3.42±0.45	-0.76±0.05	53.35±1.91	0.08±0.01	0.24±0.24	56.93±2.65	1.58±0.96	2.07±1.31
GWF+SC	73.92±4.30	18.35±3.85	24.48±4.69	52.76±0.80	3.47±1.16	-0.71±0.32	52.08±0.98	0.04±0.03	-0.12±0.08	51.93±3.58	0.61±0.83	0.58±1.19
GLCC	71.99±3.08	13.18±6.93	16.89±8.28	63.62±9.79	1.18±0.60	1.12±0.97	60.70±0.00	2.40±0.00	2.30±0.00	66.50±0.00	8.10±0.00	10.60±0.00
DCGLC	**86.70±0.87**	**42.55±3.18**	**53.02±3.10**	**81.73±0.00**	**13.57±0.00**	**18.27±0.00**	**74.93±1.22**	**18.44±1.42**	**24.72±2.32**	**66.50±2.37**	**9.16±2.09**	**11.03±2.72**

Table 1: Clustering performance (means ± std) on several graph benchmarks (%). The results highlighted in **bold** and underlined denote the best and the runner-up performance respectively, and "OM" denotes out-of-memory.

The GCL methods, e.g., GraphCL [You *et al.*, 2020] and JOAO [You *et al.*, 2021], can be applied to graph-level clustering by performing k-means [Hartigan and Wong, 1979] or SC [Ng *et al.*, 2001] on the learned representations. Compared with them, DCGLC allows graph-level representations for clustering tasks to be learned in an end-to-end manner, which guides graph representation learning with an explicitly defined clustering objective.

Some works also delved into end-to-end frameworks for jointly optimizing representation learning and clustering. For example, Xie *et al.* [2016] and Cai *et al.* [2022a] respectively investigated distance-based and subspace-based end-to-end clustering models. However, they are not committed to clustering graph-level data and considered from a single cluster perspective. Ju *et al.* [2023] proposed a graph-level contrastive clustering method for learning clusters through instance-level and cluster-level contrast. However, the cluster-level contrast in [Ju *et al.*, 2023; Li *et al.*, 2021] focuses on aligning the logits prediction from the original and augmented data, which is quite different from ours. DCGLC introduces two cluster heads to explicitly define clusters, and focuses on leveraging comprehensive cluster information from different cluster perspectives to learn more reliable cluster assignments, which is substantiated by the ablation study results in Section 4.2.

4 Experiment

In this section, we conduct a series of evaluations to demonstrate the superiority of the proposed DCGLC.

4.1 Experimental Settings

Datasets. We consider three types of graph benchmark datasets to evaluate the clustering performance, including seven molecule datasets (MUTAG, BZR, PTC-MR, PTC-MM, COX2, ER_MD, and AIDS), three biological datasets (DD, PROTEINS and ENZYMES), and two social network datasets (IMDB-BINARY and REDDIT-MULTI-5K). Details of these benchmarks refer to **Appendix A**.

Baseline Methods. We compare the proposed method with several state-of-the-art baselines, which can be broadly classified into three categories including: **1) Graph Kernel:** SP [Borgwardt and Kriegel, 2005], GK [Shervashidze *et al.*, 2009], RW [Vishwanathan *et al.*, 2010], WL [Shervashidze *et al.*, 2011], LT [Johansson *et al.*, 2014], WL-OA [Kriege *et al.*, 2016], **2) Unsupervised Graph Representation Learning:** InfoGraph [Sun *et al.*, 2020], GraphCL [You *et al.*, 2020], JOAO [You *et al.*, 2021], **3) Graph-Level Clustering:** GWF [Xu *et al.*, 2022], GLCC [Ju *et al.*, 2023]. Note that the network structure of InfoGraph [Sun *et al.*, 2020], GraphCL [You *et al.*, 2020], JOAO [You *et al.*, 2021], and GLCC [Ju *et al.*, 2023] are exactly the same as ours to guarantee a fair comparison.

Implementation Details. We employ a 4-layer GIN [Xu *et al.*, 2019] as the backbone network for the proposed DCGLC, with the aggregated dimension set to 16. We utilize an MLP-based feature projection head and cluster projection head, with the dimension of the latent layer and the clustering embedding layer both set to 10. The number of clusters is set to the number of categories in the dataset, the batch size is fixed to 64, and the training epoch is set to 300 for all datasets. We use Adam as the optimizer and follow the setting in [You *et al.*, 2021] to augment graphs automatically in each epoch. Other details and the setup of baseline models refer to **Appendix B**.

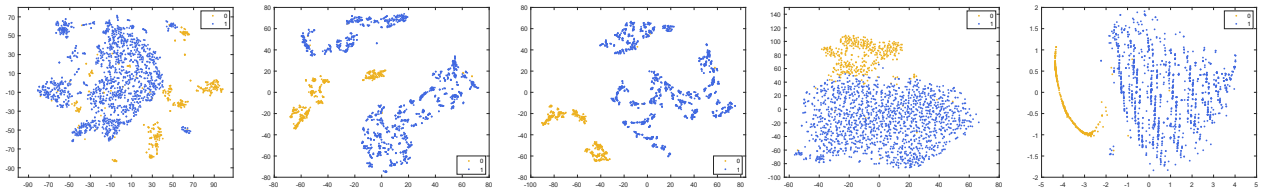
Evaluation Metrics. We use clustering accuracy (ACC), normalized mutual information (NMI), and adjusted rand index (ARI) as the evaluation metrics, and report the means and standard deviations of the three metrics based on 10 independent runs for each dataset.

4.2 Experimental Results

Comparison with State-of-the-art Baselines. Tables 1-2 present the experimental results on eight graph benchmarks from different data types (more results refer to **Appendix C**), where we compare the proposed DCGLC method with eleven state-of-the-art baselines. From this table, we have the following observations: 1) The proposed DCGLC

Method	PTC-MR			PTC-MM			COX2			ER_MD		
	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
SP	56.69±0.00	1.04±0.00	0.50±0.00	62.20±0.00	1.63±0.00	0.73±0.00	52.03±0.00	0.13±0.00	0.01±0.00	58.07±0.00	1.89±0.00	-0.79±0.00
GK	56.40±0.00	1.32±0.00	0.31±0.00	62.20±0.00	1.63±0.00	0.73±0.00	66.17±0.00	0.02±0.00	0.08±0.17	57.85±0.00	2.18±0.00	-0.91±0.00
RW	56.98±0.00	0.63±0.00	1.25±0.00	60.71±0.00	0.97±0.00	2.91±0.00	51.31±0.00	0.70±0.00	-0.92±0.00	63.68±0.00	**15.81±0.00**	6.63±0.00
WL	52.91±0.00	0.23±0.00	0.05±0.00	62.20±0.00	1.50±0.00	3.87±0.00	50.54±0.00	0.51±0.00	-0.40±0.00	61.43±0.00	2.35±0.00	2.30±0.00
LT	55.17±1.32	0.40±0.65	0.19±0.52	61.19±0.88	0.73±0.55	1.09±1.06	77.52±0.59	0.26±0.34	0.17±0.71	59.19±0.00	0.34±0.00	-0.14±0.00
WL-OA	59.30±0.00	1.77±0.00	2.95±0.00	63.39±0.00	**4.59±0.00**	2.26±0.00	50.75±0.00	0.51±0.00	-0.37±0.00	61.43±0.00	2.35±0.00	2.30±0.00
InfoGraph+KM	54.79±0.68	0.49±0.35	0.28±0.21	61.48±1.03	2.35±0.83	3.61±1.45	56.74±3.04	3.30±0.60	0.17±0.10	61.21±0.00	1.59±0.00	2.63±0.00
InfoGraph+SC	56.10±0.33	1.50±0.26	0.20±0.13	61.96±1.53	2.12±0.99	4.55±0.83	70.37±2.01	**3.56±0.99**	1.92±1.67	59.87±0.00	1.15±0.00	0.42±0.00
GraphCL+KM	54.33±0.76	1.15±0.55	0.16±0.29	58.93±0.74	0.27±0.15	0.60±0.14	68.88±0.59	1.05±0.21	0.44±0.57	60.99±0.00	1.50±0.00	3.35±0.00
GraphCL+SC	56.13±0.42	1.31±0.30	1.17±0.24	62.09±0.56	2.14±0.43	3.36±0.87	75.01±2.12	1.24±0.37	2.39±2.28	59.42±0.00	3.03±0.00	0.44±0.00
JOAO+KM	56.39±0.18	0.53±0.21	0.41±0.01	59.04±0.52	0.21±0.14	0.98±0.41	70.56±2.03	1.19±0.34	0.44±0.43	66.14±0.00	5.77±0.00	9.51±0.00
JOAO+SC	56.16±0.22	1.03±0.33	0.19±0.11	62.41±0.80	2.00±0.78	4.28±1.34	76.46±0.61	1.43±0.77	2.35±2.49	59.42±0.00	2.02±0.64	0.30±0.00
GWF+KM	56.33±3.52	1.09±0.88	1.65±1.50	53.37±3.18	0.30±0.37	0.38±1.09	57.60±4.11	1.50±0.13	2.08±1.80	51.94±1.87	0.27±0.56	-0.07±0.27
GWF+SC	55.32±4.03	0.89±0.84	1.49±1.44	53.02±1.66	0.36±0.28	0.21±0.09	58.83±4.46	1.16±0.41	1.45±1.21	59.12±0.56	0.51±0.53	-0.11±0.36
GLCC	56.10±3.29	1.21±0.63	1.38±1.15	61.61±0.24	0.63±0.41	1.24±1.38	77.37±1.11	0.02±0.03	-0.30±0.42	61.21±1.28	5.24±1.38	4.61±1.16
DCGLC	**59.98±1.11**	**2.60±0.89**	**3.66±0.92**	**65.48±0.24**	4.32±0.70	**7.34±0.15**	**78.63±0.16**	3.05±0.82	**5.42±3.15**	**66.95±0.18**	10.12±0.88	**11.29±0.25**

Table 2: Clustering performance (mean ± std) on PTC-MR/MM, COX2, and ER_MD (%). The results highlighted in **bold** and underlined denote the best and runner-up performance respectively.



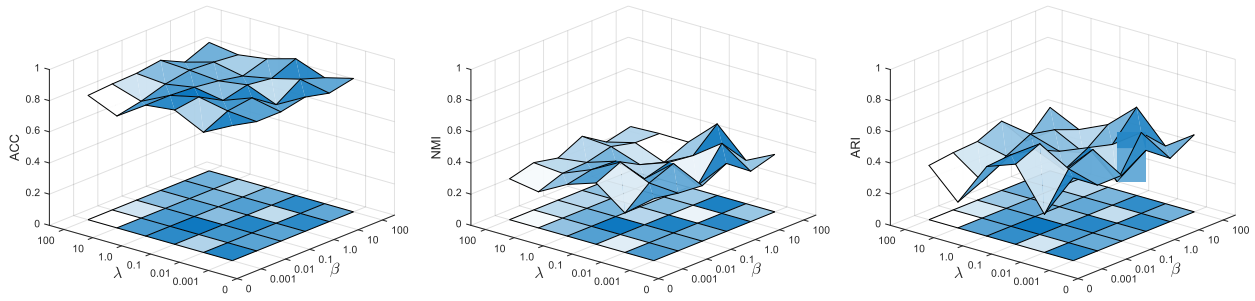
(a) InfoGraph (b) GraphCL (c) JOAO (d) GWF (e) Ours

Figure 2: Visualization of clustering embedding for the proposed DCGLC and four baseline methods on AIDS.

method demonstrates significant superiority over state-of-the-art graph kernels and graph-level clustering baselines. Particularly, compared to the GCL-based baselines, e.g., GraphCL, JOAO, and GLCC, the proposed DCGLC exhibits notable advantages across three different types of graph data. For example, on the molecule dataset MUTAG, DCGLC exceeds JOAO+KM with an improvement of 7.50%, 6.23%, and 19.28% in terms of ACC, NMI, and ARI. Besides, on the biology dataset DD, DCGLC also outperforms the runner-up GLCC by more than 14.00% across all metrics. These results clearly indicate that the dual contrastive mechanism in DCGLC can facilitate the learning of clustering-friendly graph representations. 2) Empirical results show that the one clustering perspective may not consistently yield superior performance. For instance, GWF+KM outperforms GWF+SC on PTC-MR and BZR, while the opposite is true for MUTAG and ER_MD. This phenomenon is also observed in other state-of-the-art baselines, e.g., GraphCL, and JOAO. These observations suggest that different clustering perspectives reveal distinct information for clustering, which may exhibit different validity for various graph data. Besides, this also highlights the rationale for leveraging the cluster information from multiple clustering perspectives to promote graph-level clustering.

Visualization of Clustering Embedding. We employ t-SNE [Van der Maaten and Hinton, 2008] to visualize the clustering embeddings to facilitate a more intuitive comparison. As shown in Figure 2, it is evident that InfoGraph yields a disorganized result with no discernible clusters. While GraphCL and JOAO demonstrate better performance compared to InfoGraph, the samples of the same color are still widely dispersed. Besides, although GWF shows clearer boundaries between classes, the gap between the two classes is small, resulting in blurred clustering boundaries. Compared with them, the clustering structures of DCGLC are more compact, with different clusters more separated and samples from the same cluster being significantly closer in proximity. The improved compactness and separability of the clusters indicate the effectiveness of DCGLC in capturing and leveraging the underlying structure and relationships within the graph data for more accurate clustering.

Impact of λ and β. We evaluate the impact of the variation of two hyper-parameters, λ, and β, on the clustering performance. Specifically, we conduct the parameter analysis by intentionally setting the values of λ and β within a broad range of $[0.001, 100]$. The results depicted in Figure 3 show the performance variations observed on MUTAG (more results refer to **Appendix D**). The empirical results reveal that excessively high values of λ generally lead to a negative impact on clustering performance. For instance, when λ is set to 100, the performance becomes unsatisfactory. Interestingly, our method demonstrates a remarkable insensitivity to changes in the value of β, as evidenced by its consistently stable performance across different values. Notably, the ARI tends to be more sensitive to variations in both λ and β. Overall, these findings collectively underscore the robustness and stability



(a) ACC (b) NMI (c) ARI

Figure 3: The impact of λ and β to the clustering performance on MUTAG. λ and β changes in the range of $[0.001, 100]$.

of DCGLC across a broad range of λ and β values.

Ablation Study. We conduct an ablation study to demonstrate the effectiveness of the proposed method and the rationale of integrating the cluster information from different perspectives. We construct three degradation variants for the proposed DCGLC, including:

1) **DCGLC_v1:** Drop out the cluster-perspective contrast component, and only consider the Euclidian-based clustering head and graph-level features contrast.
2) **DCGLC_v2:** Drop out the cluster-perspective contrast component, and only consider the subspace-based clustering head and graph-level features contrast.
3) **DCGLC_v3:** Drop out the graph-level features contrast component and only train under the guidance of cluster-perspective contrast.

Dataset	Metrics	DCGLC_v1	DCGLC_v2	DCGLC_v3	DCGLC
MUTAG	ACC	78.94±5.64	78.99±3.11	84.47±1.48	**86.70±0.87**
	NMI	26.04±12.07	23.85±5.42	41.62±2.11	**42.55±3.18**
	ARI	32.85±14.16	32.95±6.86	47.17±5.17	**53.02±3.10**
BZR	ACC	79.48±1.11	80.74±0.80	79.16±0.79	**81.73±0.00**
	NMI	5.78±4.32	9.31±3.01	8.05±1.62	**13.57±0.00**
	ARI	11.41±5.46	12.35±4.68	15.24±4.66	**18.27±0.00**
AIDS	ACC	89.96±2.71	80.00±0.00	88.55±0.66	**96.77±0.33**
	NMI	50.19±10.18	12.22±2.04	38.96±2.69	**73.51±2.30**
	ARI	56.00±13.71	7.77±6.57	46.14±3.28	**85.74±1.45**

Table 3: Experimental results of ablation study on three datasets.

In Table 3, we thoroughly evaluate DCGLC in comparison to all degradation variants on three datasets (more results refer to **Appendix E**), revealing several key insights: 1) The Euclidean-based and subspace-based cluster perspectives each possess their own set of strengths and weaknesses. The former demonstrates superior performance on MUTAG, whereas the latter outperforms on BZR. This highlights the importance of considering different clustering perspectives, as their effectiveness can vary depending on the characteristics of graph data. 2) Relying solely on a single cluster perspective is insufficient for capturing comprehensive cluster information, as we observe consistent performance degradation DCGLC_v1 and DCGLC_v2. The results provide strong evidence for the validity of incorporating cluster information from different cluster perspectives in graph-level clustering. 3) The importance of graph-level features contrast is demonstrated from the results of DCGLC_v3. Notably, removing this component leads to noticeable performance degradation, reaffirming its indispensable role in enabling the network to learn more effective graph representations.

5 Conclusion

In this paper, we proposed dual contrastive graph-level clustering (DCGLC). DCGLC leverages graph contrastive learning and introduces an explicit clustering objective to learn discriminative graph-level representations that are conducive to clustering. We highlight the importance of capturing comprehensive cluster information from multiple cluster perspectives to graph-level clustering, and propose a contrastive mechanism to align cluster information from different cluster perspectives. DCGLC is an end-to-end framework that integrates graph contrastive learning and cluster-perspective contrast, leading to a mutual enhancement during training. Extensive experiments on several popular graph benchmarks compared with state-of-the-art methods fully demonstrate the superiority of DCGLC. The main limitation of this work stems from the exploration of the graph augmentation mechanism beneficial for capturing better clustering structures, which is a promising research issue for future work.

Acknowledgments

This research is supported by the National Natural Science Foundation of China (Grants No.U21A20472, No.62276065, and No.62376236), and Shenzhen Research Institute of Big Data (Grant No.T00120210002).

Additionally, this research is supported by the National Research Foundation Singapore and DSO National Laboratories under the AI Singapore Programme (AISG Award No: AISG2-RP-2020-018). Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not reflect the views of National Research Foundation, Singapore.

Contribution Statement

Jinyu Cai and Yunhe Zhang contributed equally to this work as co-first authors.

References

[Agarwal, 2006] Shivani Agarwal. Ranking on graph data. In *ICML*, pages 25–32, 2006.

[Aggarwal, 2011] Charu C Aggarwal. *An introduction to social network data analytics*. Springer, 2011.

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

[Borgwardt and Kriegel, 2005] Karsten M Borgwardt and Hans-Peter Kriegel. Shortest-path kernels on graphs. In *ICDM*, pages 8–pp. IEEE, 2005.

[Cai *et al.*, 2022a] Jinyu Cai, Jicong Fan, Wenzhong Guo, Shiping Wang, Yunhe Zhang, and Zhao Zhang. Efficient deep embedded subspace clustering. In *CVPR*, pages 1–10, 2022.

[Cai *et al.*, 2022b] Jinyu Cai, Shiping Wang, Chaoyang Xu, and Wenzhong Guo. Unsupervised deep clustering via contractive feature representation and focal loss. *Pattern Recognition*, 123:108386, 2022.

[Cai *et al.*, 2024a] Jinyu Cai, Yi Han, Wenzhong Guo, and Jicong Fan. Deep graph-level clustering using pseudo-label-guided mutual information maximization network. *NCAA*, pages 1–16, 2024.

[Cai *et al.*, 2024b] Jinyu Cai, Yunhe Zhang, Shiping Wang, Jicong Fan, and Wenzhong Guo. Wasserstein embedding learning for deep clustering: A generative approach. *IEEE T-MM*, 2024.

[Chen *et al.*, 2022] Man-Sheng Chen, Chang-Dong Wang, Dong Huang, Jian-Huang Lai, and Philip S. Yu. Efficient orthogonal multi-view subspace clustering. In *SIGKDD*, pages 127–135. ACM, 2022.

[Chen *et al.*, 2023a] Man-Sheng Chen, Chang-Dong Wang, and Jian-Huang Lai. Low-rank tensor based proximity learning for multi-view clustering. *IEEE T-KDE*, 35(5):5076–5090, 2023.

[Chen *et al.*, 2023b] Zhaoliang Chen, Lele Fu, Shunxin Xiao, Shiping Wang, Claudia Plant, and Wenzhong Guo. Multi-view graph convolutional networks with differentiable node selection. *ACM T-KDD*, 18(1):1–21, 2023.

[Chen *et al.*, 2023c] Zhaoliang Chen, Lele Fu, Jie Yao, Wenzhong Guo, Claudia Plant, and Shiping Wang. Learnable graph convolutional network and feature fusion for multi-view learning. *Information Fusion*, 95:109–119, 2023.

[Fan *et al.*, 2022] Jicong Fan, Yiheng Tu, Zhao Zhang, Mingbo Zhao, and Haijun Zhang. A simple approach to automated spectral clustering. In *NeurIPS*, pages 9907–9921, 2022.

[Fu *et al.*, 2022] Lele Fu, Zhaoliang Chen, Yongyong Chen, and Shiping Wang. Unified low-rank tensor learning and spectral embedding for multi-view subspace clustering. *IEEE T-MM*, 2022.

[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.

[Johansson *et al.*, 2014] Fredrik Johansson, Vinay Jethava, Devdatt Dubhashi, and Chiranjib Bhattacharyya. Global graph kernels using geometric embeddings. In *ICML*, pages 694–702. PMLR, 2014.

[Ju *et al.*, 2023] Wei Ju, Yiyang Gu, Binqi Chen, Gongbo Sun, Yifang Qin, Xingyuming Liu, Xiao Luo, and Ming Zhang. Glcc: A general framework for graph-level clustering. In *AAAI*, volume 37, pages 4391–4399, 2023.

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

[Kriege *et al.*, 2016] Nils M Kriege, Pierre-Louis Giscard, and Richard Wilson. On valid optimal assignment kernels and applications to graph classification. *NeurIPS*, 29, 2016.

[Li *et al.*, 2021] Yunfan Li, Peng Hu, Zitao Liu, Dezhong Peng, Joey Tianyi Zhou, and Xi Peng. Contrastive clustering. In *AAAI*, volume 35, pages 8547–8555, 2021.

[Li *et al.*, 2023] Xingfeng Li, Yinghui Sun, Quansen Sun, Zhenwen Ren, and Yuan Sun. Cross-view graph matching guided anchor alignment for incomplete multi-view clustering. *Information Fusion*, 100:101941, 2023.

[Liu *et al.*, 2022] Yue Liu, Wenxuan Tu, Sihang Zhou, Xinwang Liu, Linxuan Song, Xihong Yang, and En Zhu. Deep graph clustering via dual correlation reduction. In *AAAI*, volume 36, pages 7603–7611, 2022.

[Liu *et al.*, 2023a] Yue Liu, Xihong Yang, Sihang Zhou, and Xinwang Liu. Simple contrastive graph clustering. *IEEE T-NNLS*, 2023.

[Liu *et al.*, 2023b] Yue Liu, Xihong Yang, Sihang Zhou, Xinwang Liu, Zhen Wang, Ke Liang, Wenxuan Tu, Liang Li, Jingcan Duan, and Cancan Chen. Hard sample aware network for contrastive deep graph clustering. In *AAAI*, volume 37, pages 8914–8922, 2023.

[Ng *et al.*, 2001] Andrew Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. *NeurIPS*, 14, 2001.

[Pan and Kang, 2021] Erlin Pan and Zhao Kang. Multi-view contrastive graph clustering. *NeurIPS*, 34:2148–2159, 2021.

[Park and Neville, 2019] Hogun Park and Jennifer Neville. Exploiting interaction links for node classification with deep graph neural networks. In *IJCAI*, volume 2019, pages 3223–3230, 2019.

[Rong *et al.*, 2020] Yu Rong, Yatao Bian, Tingyang Xu, Weiyang Xie, Ying Wei, Wenbing Huang, and Junzhou Huang. Self-supervised graph transformer on large-scale molecular data. *NeurIPS*, 33:12559–12571, 2020.

[Schaeffer, 2007] Satu Elisa Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27–64, 2007.

[Shervashidze *et al.*, 2009] Nino Shervashidze, SVN Vishwanathan, Tobias Petri, Kurt Mehlhorn, and Karsten Borgwardt. Efficient graphlet kernels for large graph comparison. In *AISTATS*, pages 488–495. PMLR, 2009.

[Shervashidze *et al.*, 2011] Nino Shervashidze, Pascal Schweitzer, Erik Jan Van Leeuwen, Kurt Mehlhorn, and Karsten M Borgwardt. Weisfeiler-lehman graph kernels. *JMLR*, 12(9), 2011.

[Sohn, 2016] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. *NeurIPS*, 29, 2016.

[Sun and Fan, 2024] Yan Sun and Jicong Fan. Mmd graph kernel: Effective metric learning for graphs via maximum mean discrepancy. In *ICLR*, 2024.

[Sun *et al.*, 2020] Fan-Yun Sun, Jordon Hoffman, Vikas Verma, and Jian Tang. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. In *ICLR*, 2020.

[Sun *et al.*, 2024] Ziheng Sun, Chris Ding, and Jicong Fan. Lovász principle for unsupervised graph representation learning. *NeurIPS*, 36, 2024.

[Tu *et al.*, 2021] Wenxuan Tu, Sihang Zhou, Xinwang Liu, Xifeng Guo, Zhiping Cai, En Zhu, and Jieren Cheng. Deep fusion clustering network. In *AAAI*, volume 35, pages 9978–9987, 2021.

[Van der Maaten and Hinton, 2008] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 9(11):2579–2605, 2008.

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

[Vishwanathan *et al.*, 2010] S Vichy N Vishwanathan, Nicol N Schraudolph, Risi Kondor, and Karsten M Borgwardt. Graph kernels. *JMLR*, 11:1201–1242, 2010.

[Wang *et al.*, 2017] Chun Wang, Shirui Pan, Guodong Long, Xingquan Zhu, and Jing Jiang. Mgae: Marginalized graph autoencoder for graph clustering. In *CIKM*, pages 889–898, 2017.

[Wu *et al.*, 2020] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE T-NNLS*, 32(1):4–24, 2020.

[Wu *et al.*, 2024a] Zhihao Wu, Zhaoliang Chen, Shide Du, Sujia Huang, and Shiping Wang. Graph convolutional network with elastic topology. *Pattern Recognition*, page 110364, 2024.

[Wu *et al.*, 2024b] Zhihao Wu, Zhao Zhang, and Jicong Fan. Graph convolutional kernel machine versus graph convolutional networks. *NeurIPS*, 36, 2024.

[Xia *et al.*, 2022] Wei Xia, Qianqian Wang, Quanxue Gao, Ming Yang, and Xinbo Gao. Self-consistent contrastive attributed graph clustering with pseudo-label prompt. *IEEE T-MM*, 2022.

[Xie *et al.*, 2016] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *ICML*, pages 478–487. PMLR, 2016.

[Xu *et al.*, 2019] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *ICLR*, 2019.

[Xu *et al.*, 2022] Hongteng Xu, Jiachang Liu, Dixin Luo, and Lawrence Carin. Representing graphs via gromov-wasserstein factorization. *IEEE T-PAMI*, 2022.

[Yang *et al.*, 2016] Jianwei Yang, Devi Parikh, and Dhruv Batra. Joint unsupervised learning of deep representations and image clusters. In *CVPR*, pages 5147–5156, 2016.

[Yang *et al.*, 2023] Zhenyu Yang, Ge Zhang, Jia Wu, Jian Yang, Quan Z Sheng, Shan Xue, Chuan Zhou, Charu Aggarwal, Hao Peng, Wenbin Hu, et al. A comprehensive survey of graph-level learning. *arXiv preprint arXiv:2301.05860*, 2023.

[Yao *et al.*, 2023] Yinghua Yao, Yuangang Pan, Jing Li, Ivor W Tsang, and Xin Yao. Sanitized clustering against confounding bias. *Machine Learning*, pages 1–20, 2023.

[You *et al.*, 2020] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. *NeurIPS*, 33:5812–5823, 2020.

[You *et al.*, 2021] Yuning You, Tianlong Chen, Yang Shen, and Zhangyang Wang. Graph contrastive learning automated. In *ICML*, pages 12121–12132. PMLR, 2021.

[Zhang *et al.*, 2022] Chao Zhang, Huaxiong Li, Caihua Chen, Xiuyi Jia, and Chunlin Chen. Low-rank tensor regularized views recovery for incomplete multiview clustering. *IEEE T-NNLS*, 2022.

[Zhang *et al.*, 2023] Chao Zhang, Huaxiong Li, Wei Lv, Zizheng Huang, Yang Gao, and Chunlin Chen. Enhanced tensor low-rank and sparse representation recovery for incomplete multi-view clustering. In *AAAI*, pages 11174–11182, 2023.

[Zhao *et al.*, 2021] Han Zhao, Xu Yang, Zhenru Wang, Erkun Yang, and Cheng Deng. Graph debiased contrastive learning with joint representation clustering. In *IJCAI*, pages 3434–3440, 2021.

[Zhou *et al.*, 2020] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020.

[Zhu *et al.*, 2021a] Yanqiao Zhu, Yichen Xu, Qiang Liu, and Shu Wu. An Empirical Study of Graph Contrastive Learning. In *NeurIPS Systems Track on Datasets and Benchmarks*, volume 1, 2021.

[Zhu *et al.*, 2021b] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Graph Contrastive Learning with Adaptive Augmentation. In *WWW*, pages 2069–2080, 2021.

Appendix for Dual Contrastive Graph-Level Clustering with Multiple Cluster Perspectives Alignment

Jinyu Cai[1*], **Yunhe Zhang**[2,3*], **Jicong Fan**[2,3], **Yali Du**[4] and **Wenzhong Guo**[5†]
[1]Institute of Data Science, National University of Singapore, Singapore
[2]Shenzhen Research Institute of Big Data, Shenzhen, China
[3]School of Data Science, The Chinese University of Hong Kong, Shenzhen, China
[4]Department of Informatics, King's College London, United Kingdom
[5]College of Computer and Data Science, Fuzhou University, China
jinyucai@nus.edu.sg, zhangyhannie@gmail.com, fanjicong@cuhk.edu.cn, yali.du@kcl.ac.uk, guowenzhong@fzu.edu.cn

Abstract

This appendix includes the following contents:

1. Details of the graph benchmarks used in the experiment.
2. The detailed experimental settings including the network architecture, trade-off parameters, training details, baseline settings, etc.
3. More experimental results on the remaining datasets, e.g., AIDS, PROTEINS, etc.
4. More parameter analysis of λ and β.
5. More ablation study results.

A Appendix A: Details of Graph Benchmarks

The graph benchmarks used in this paper source from TUDataset [Morris *et al.*, 2020] [1], a publicly available database. Here, we supplement more details for all the graph benchmarks in Table 1, including the total number of graphs, the average number of nodes and edges, the classes of nodes and graphs, and the type of data.

B Appendix B: Experimental Settings

We supplement more details of the experimental settings in the paper, including the network structure, trade-off parameter settings, training details, baseline settings, etc.

- **Network structure:** We employ a 4-layer GIN [Xu *et al.*, 2019] as the backbone network for our method, with the aggregated dimension set to 16. we utilize an MLP-based feature projection head and cluster projection head in our method, with the dimension of the latent layer and the clustering embedding layer both set to 10. The number of clusters is set to be the same as the number of categories in the dataset. The source code is also included in the supplementary material for additional details and reproducibility.
- **Trade-off parameter settings:** There are two trade-off parameters in our method, i.e., λ, and β. We vary their values in the range of $[0.001, 0.01, 0.1, 1, 10, 100]$, and adopt the hyper-parameter grid search strategy to find the optimal settings. Similar strategy is also applied to other methods if applicable for fairness.
- **Training details:** We fix the batch size as 64 to the experiment on each experiment. We train the model for 300 epochs on all datasets with Adam [Kingma and Ba, 2014] as the optimizer. Note that for the graph augmentation in each epoch, we follow the setting in [You *et al.*, 2021] to augment graphs automatically.
- **Baseline settings:** For graph kernels, we evaluate their clustering performance by SC [Ng *et al.*, 2001], since they generally learn similarity between graphs, which can naturally be related to spectral clustering. For unsupervised graph representation learning approaches, we evaluate their performance by both k-means [Hartigan and Wong, 1979] and SC [Ng *et al.*, 2001]. We run experiments for each of them based on the officially released codes. Particularly, for the graph-level clustering baseline GLCC [Ju *et al.*, 2023], we report the performance directly from its original paper for DD and IMDB-BINARY, and reproduce its code to conduct experiments on the remaining benchmarks. Regarding GWF [Xu *et al.*, 2022], the released code only includes the k-means evaluation, therefore we reproduce its results on SC following its default settings.
- **Implementation:** Our implementation is based on PyTorch Geometric [Fey and Lenssen, 2019] library, and all experiments are run on NVIDIA Tesla A100 GPU with AMD EPYC 7532 CPU.

C Appendix C: More Experimental Results

In this section, we supplement additional experimental results for the remaining datasets in Table 2. From this table, we can observe that the proposed DCGLC method consistently achieves the best clustering results in terms of ACC, NMI, and ARI in the majority of graph benchmarks, and surpasses state-of-the-art baselines such as WL-OA, JOAO,

*Co-first author.
†Corresponding author.
[1]https://chrsmrrs.github.io/datasets/docs/datasets/

Dataset name	#Graphs	#Average nodes	#Average edges	#Node classes	#Graph classes	# Data Type
MUTAG	188	17.93	19.79	7	2	Molecule
BZR	405	35.75	38.36	10	2	Molecule
PTC_MR	344	14.29	14.69	18	2	Molecule
PTC_MM	336	13.97	14.32	20	2	Molecule
COX2	467	41.22	43.45	8	2	Molecule
ER_MD	446	21.33	234.85	10	2	Molecule
AIDS	2,000	15.69	16.20	38	2	Molecule
DD	1,178	284.32	715.66	82	2	Biology
PROTEINS	1,113	39.06	72.82	3	2	Biology
ENZYMES	600	32.63	62.14	3	6	Biology
IMDB-Binary	1,000	19.77	96.53	–	2	Social networks
REDDIT-MULTI-5K	4,999	508.52	594.87	–	5	Social networks

Table 1: Detailed information of the datasets used in the experiment.

Method	AIDS			PROTEINS			ENZYMES			REDDIT-MULTI-5K		
	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
SP	79.49±0.84	0.39±0.62	-0.71±1.13	64.42±0.00	6.03±0.00	5.87±0.00	22.00±0.00	2.57±0.00	1.69±0.00	20.02±0.00	0.05±0.00	0.00±0.00
GK	79.95±0.00	0.04±0.00	-0.07±0.00	59.61±0.22	0.24±0.18	0.10±0.19	17.07±0.13	0.80±0.25	0.00±0.00	OM	OM	OM
RW	79.90±0.00	0.09±0.00	-0.15±0.00	OM	OM	OM	17.00±0.00	0.66±0.00	0.25±0.00	OM	OM	OM
WL	78.50±0.00	1.17±0.00	-2.09±0.00	60.38±0.00	1.55±0.00	0.81±0.00	21.00±0.00	3.09±0.00	1.48±0.00	20.00±0.00	0.00±0.00	0.00±0.00
LT	79.95±0.00	0.04±0.00	-0.07±0.00	OM	OM	OM	17.00±0.09	0.42±0.11	0.00±0.00	OM	OM	OM
WL-OA	80.40±0.00	2.46±0.00	2.38±0.00	60.38±0.00	1.55±0.00	0.81±0.00	20.00±0.00	1.35±0.00	0.32±0.00	OM	OM	OM
InfoGraph+KM	92.21±0.81	54.49±3.53	63.78±3.84	59.22±0.21	3.22±1.94	0.00±0.00	22.06±0.98	2.40±0.45	1.25±0.52	20.16±0.02	0.30±0.05	0.00±0.00
InfoGraph+SC	95.65±1.55	72.21±9.20	80.17±7.19	64.02±2.31	5.17±1.87	7.06±2.65	23.75±0.50	4.64±0.65	2.23±0.41	20.00±0.00	0.00±0.00	0.00±0.00
GraphCL+KM	90.40±1.06	46.56±4.31	55.29±5.28	59.47±0.01	0.37±0.31	0.00±0.00	21.50±0.22	1.55±0.12	0.90±0.09	20.32±0.00	0.56±0.00	0.00±0.00
GraphCL+SC	96.08±1.96	72.97±10.86	81.65±8.51	59.96±0.10	2.81±0.07	3.88±0.08	25.28±0.28	4.75±0.36	2.03±0.26	20.08±0.00	0.16±0.00	0.00±0.00
JOAO+KM	88.25±0.00	38.02±0.00	44.62±0.00	59.48±0.00	0.64±0.05	-0.06±0.00	21.66±0.37	1.60±0.01	0.94±0.02	20.34±0.00	0.60±0.00	0.00±0.00
JOAO+SC	80.13±0.02	0.84±0.15	0.80±0.14	59.75±0.00	0.47±0.00	0.17±0.00	24.65±0.44	4.85±0.37	2.07±0.18	20.39±0.49	0.08±0.00	0.01±0.01
GWF+KM	96.43±1.71	74.48±9.15	84.71±7.02	66.87±2.36	9.07±1.21	11.43±3.19	**28.55±0.20**	6.02±0.55	3.16±0.20	OM	OM	OM
GWF+SC	96.44±2.92	**76.01±15.23**	83.54±13.61	68.79±2.05	10.17±1.74	13.88±2.72	25.66±1.57	5.24±1.28	1.78±0.61	OM	OM	OM
GLCC	79.02±0.62	4.18±2.01	5.05±2.13	60.65±2.69	2.08±1.43	4.16±2.28	19.89±1.09	2.42±0.18	0.19±0.12	23.50±0.48	6.57±3.56	4.00±0.80
DCGLC	**96.77±0.33**	73.51±2.30	**85.74±1.45**	**68.89±2.04**	**10.90±1.35**	**14.32±2.88**	28.43±1.28	**6.57±0.20**	**3.78±0.47**	**33.24±2.34**	**8.81±2.28**	**7.16±1.67**

Table 2: Clustering performance (mean ± std) on AIDS, PROTEINS, ENZYMES, and REDDIT-MULTI-5K (%). The results highlighted in **bold** and underlined denote the best and runner-up performance respectively, and "OM" denotes out-of-memory.

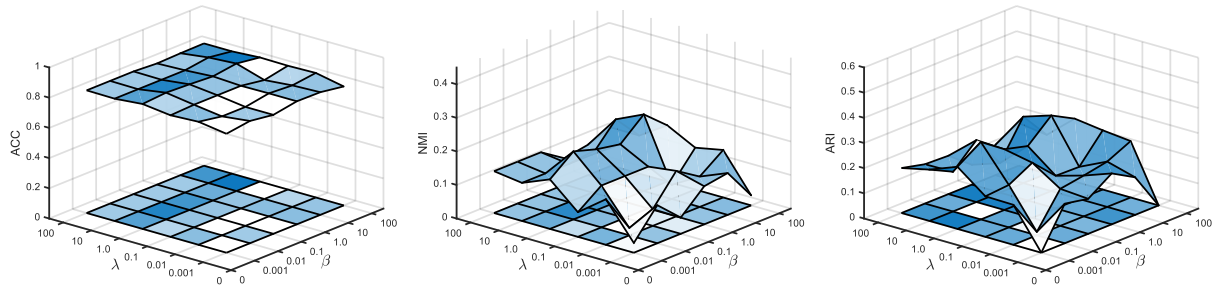
GWF, and GLCC. Although DCGLC obtains runner-up performance in very few cases, it still shows competitive results compared with the best one. More importantly, unlike other baselines that excel on only certain datasets, DCGLC demonstrates significant superiority in all datasets. These observations strongly highlight the exceptional clustering performance and adaptability of the proposed DCGLC method across various types of graph data.

D Appendix D: More Analysis of λ and β

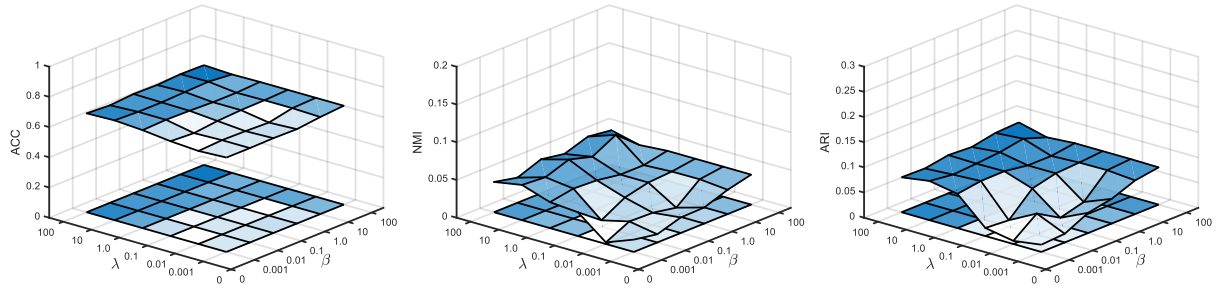
In this section, we supplement more analysis of the impact of hyper-parameters λ and β on the graph-level clustering performance. The experimental configurations remain consistent with those detailed in Appendix A. As illustrated in Figure 1, we show the experimental results on BZR and PTC-MM, where we can intuitively evaluate the performance of DCGLC by observing the shades of colors in the box at the bottom. The figure demonstrates that DCGLC consistently maintains robust clustering performance within specific intervals of λ and β values. It is worth noting that fluctuations in clustering performance for certain values are a consequence of the broad spectrum of values tested for these hyper-parameters (from 0.001 to 100). Importantly, as the values of λ and β are set to close to zero, a stark clustering performance decline becomes evident. This phenomenon reveals the essential role of the clustering objective, which enables the joint optimization of clustering and graph representation learning. Furthermore, it also highlights the significance of the cluster perspectives contrast mechanism, which facilitates the integration of cluster information from diverse perspectives. Overall, the experiment supports the robustness of the proposed method, and desirable clustering performance can be achieved within a relatively wide range of values of λ and β.

E Appendix E: More Ablation Study Results

In this section, we supplement more ablation study results to further substantiate the effectiveness of each component within the proposed DCGLC method. Table 3 summarizes the ablation study results on PTC-MM and PTC-MR. We can observe that the clustering performance of all three variants of DCGLC decreases in most cases, which is consistent with the observations in the main text. For instance, the comparison of DCGLC_v1 and DCGLC_v2 demonstrates the validity of different cluster perspectives. The performance decline in DCGLC_v3 compared with DCGLC also highlights the importance of graph representation learning. Overall, the ablation study points out that the proposed DCGLC method advances graph-level clustering by leveraging the cluster infor-



(a) BZR



(b) PTC_MM

Figure 1: Parameter sensitivity analysis on BZR and PTC_MM, where λ and β changes in the range of $[0.001, 100]$.

Dataset	Metrics	DCGLC_v1	DCGLC_v2	DCGLC_v3	DCGLC
PTC-MM	ACC	65.28±0.85	63.54±0.76	60.27±1.34	**65.48±0.24**
	NMI	3.93±0.77	2.58±0.48	1.88±0.33	**4.32±0.70**
	ARI	**7.49±1.17**	4.55±1.42	2.94±0.34	7.34±0.15
PTC-MR	ACC	57.50±1.34	58.34±1.21	57.27±1.16	**59.98±1.11**
	NMI	1.35±0.77	1.90±0.70	1.23±0.28	**2.60±0.89**
	ARI	1.56±0.81	2.27±0.88	1.78±0.58	**3.66±0.92**

Table 3: More ablation study results on PTC-MM and PTC-MR.

mation from multiple cluster perspectives and an end-to-end framework designed for the joint optimization of graph contrastive learning and clustering.

References

[Fey and Lenssen, 2019] Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428*, 2019.

[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.

[Ju *et al.*, 2023] Wei Ju, Yiyang Gu, Binqi Chen, Gongbo Sun, Yifang Qin, Xingyuming Liu, Xiao Luo, and Ming Zhang. Glcc: A general framework for graph-level clustering. In *AAAI*, volume 37, pages 4391–4399, 2023.

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

[Morris *et al.*, 2020] Christopher Morris, Nils M. Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. Tudataset: A collection of benchmark datasets for learning with graphs. In *ICML Workshop on Graph Representation Learning and Beyond*, 2020.

[Ng *et al.*, 2001] Andrew Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. *NeurIPS*, 14, 2001.

[Xu *et al.*, 2019] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *ICLR*, 2019.

[Xu *et al.*, 2022] Hongteng Xu, Jiachang Liu, Dixin Luo, and Lawrence Carin. Representing graphs via gromov-wasserstein factorization. *IEEE T-PAMI*, 2022.

[You *et al.*, 2021] Yuning You, Tianlong Chen, Yang Shen, and Zhangyang Wang. Graph contrastive learning automated. In *ICML*, pages 12121–12132. PMLR, 2021.