Improving Graph Clustering with Multi-Granularity Debiased Contrastive Learning

Anonymous ACL submission

Abstract

Recently, deep graph clustering achieves significant success by utilizing both the node attribute features and the graph structure information. However, the existing methods still have some limitations: (1) lack of a flexible mechanism to fuse multi-granularity information learned from different views. (2) introduce the noise positive-negative sample pairs lead to reduced the model performance. To tackle these problems, we propose a debiased contrastive learning framework DCL-MGI, which integrates the multi-granularity information of graph data. Specifically, two contrastive learning modules are constructed to capture multi-granularity feature information from node-level and graph-level, respectively. Meanwhile, an adaptive strategy of fusing stable graph structure information and node representations is proposed to select unbiased contrastive sample pairs, which reduces the false-negative samples. Furthermore, we utilize the temporal entropy metric to evaluate the sample quality under each view and communicate the two independent contrastive learning modules in a collaborative training manner. Experimental results on six real-world datasets demonstrate that our proposed framework enhances state-of-the-art methods on the graph clustering task.

1 Introduction

Graph clustering is a fundamental data analysis task dividing similar samples into the same cluster while separating dissimilar ones. Recently, numerous deep graph clustering methods have been proposed and applied in many scenarios, such as traffic flow forecast (Guo et al., 2021) and signal propagation (Huang et al., 2020; Jia et al., 2020). According to the learning objective, deep graph clustering can be divided into reconstruction-based methods and contrastive-based methods.

For the reconstruction-based methods, most of them utilize Graph Convolutional Networks (GCNs) and Auto-Encoder (AE) to encode both the graph structure information and node attribute features. For example, (Kipf and Welling, 2016) propose the Graph Auto-Encoder (GAE) and its variant. (Bo et al., 2020) propose Structural Deep Clustering Network (SDCN) that jointly learns GAE and AE in a uniform framework. In addition, (Wang et al., 2019) and (Peng et al., 2021) introduce the attention mechanism for graph clustering. Although reconstruction-based methods can learn node representations without labeled data, the above methods ignore the local and global information of the graph.

Another group of methods is regarded as contrastive-based methods. The key to contrastive learning is to maximize the similarity of positive pairs and minimize that of negative pairs. Specifically, (Hassani and Khasahmadi, 2020) randomly sample nodes and edges from different views. Further, (Zhao et al., 2021) construct the node clustering labels to select negative samples and (Pan and Kang, 2021) utilize k-nearest neighbors to select positive samples. The aforementioned methods have achieved preliminary success. However, the above methods construct contrastive sample pairs randomly or entirely rely on node representations, which will bring noise positive-negative samples. This phenomenon is named as **sampling bias** (Chuang et al., 2020).

To address these issues, we propose a novel **D**ebiased **C**ontrastive **L**earning framework based on **M**ulti-**G**ranularity feature **I**nteraction (DCL-MGI). First, to capture local node features and global distribution of clusters, DCL-MGI learns clustering-oriented node representations by two individual contrastive learning modules. Then, an adaptive fusion strategy is developed for selecting unbiased contrastive sample pairs that dynamically integrates the node features and the graph structure information. Further, to interact with multi-granularity feature information, a sample quality evaluation metric based on training dynamics and

information entropy is proposed and the two individual contrastive learning modules are jointly optimized by exchanging hard sample sets. Finally, the suitability of the contrastive learning objective on the graph clustering task is formally analyzed.

Our contributions can be summarized as follows:

- DCL-MGI fuses multi-granularity graph information in a unified framework, alleviating the objective mismatch and sampling bias.
- A temporal entropy-based sample evaluation metric is developed. Using this metric, two independent contrastive learning models can interact with each other effectively.
- Extensive experiments demonstrate the effectiveness of DCL-MGI against state-of-the-art methods on the graph clustering task.

2 Related Work

2.1 Contrastive Learning

As an unsupervised representation learning manner, contrastive learning has achieved impressive performances in many downstream tasks. For each target sample (also name as anchor), contrastive learning aims to capture the similarity with positive samples while expanding the dissimilarity with negative samples (Hadsell et al., 2006). Following this principle, several classical loss functions have been proposed. Specifically, (Chopra et al., 2005) design a triplet loss to capture the similarity between target space and input space. (Gutmann and Hyvärinen, 2010) propose the noise contrastive estimation (NCE) loss. Further, (Oord et al., 2018) propose the InfoNCE which is widely utilized. (Chen et al., 2020) adopt the normalized temperature-scaled cross-entropy loss (NT-Xent) to identify positive sample pairs. The above loss functions have been widely applied in many fields, including NLP (Sun et al., 2020; Kong et al., 2019), recommendation (Wu et al., 2021) and CV (Li et al., 2021).

2.2 Graph Clustering

In recent years, several GCN-based methods are designed for graph clustering. In general, existing methods can be divided into reconstruction-based methods and contrastive-based methods. For reconstruction-based methods, most of them utilize the AE framework to learn reconstruction loss function. Specifically, (Kipf and Welling, 2016) propose the GAE and VGAE, which merge GCN as the encoder into the AE framework. (Wang et al., 2019) utilize attention mechanism to identify the importance of neighboring nodes, and supervise the training process by KL-divergence. (Pan et al., 2019) employ the adversarial training principle to learn the node representations. (Bo et al., 2020) integrate the structure information into deep clustering and utilize a dual self-supervised mechanism to unify AE and GCN. (Peng et al., 2021) exploit attention mechanism to integrate node attribute feature and graph topological information. For contrastive-based methods, the learning objective function is designed by constructing positive and negative pairs. For example, (Hassani and Khasahmadi, 2020) design the multi-view graph representation learning method (MVGRL) to integrate graph information from multi-views. (Zhao et al., 2021) propose the graph debiased contrastive learning framework (GDCL) to jointly learn graph representations and clustering results. Meanwhile, GDCL develops a debiased sampling strategy to decrease the false-negative samples.

To combine the above two categories of methods, DCL-MGI selects reconstruction-based methods as backbones and adopts a clustering-oriented contrastive learning loss. DCL-MGI can capture multi-scale information from graph. Unlike the exiting methods, our methods focus on the training sample quality under different views and realizes multi-granularity feature information interaction in an collaborative training manner. More importantly, graph structure information is utilized to intervene the sampling process in contrastive learning, which decrease the false positive-negative sample pairs.

3 Preliminaries

Given the graph as $G = \{V, E, X\}$. $V = \{v_i\}_{i=1}^{n}$ is the set of n nodes. E indicates the adjacency relationships (i.e., edges) between node pairs. In general, E can be transformed to $A \in \mathbb{R}^{n \times n}$, where $e_{ij} \in E$ is equivalent to $A_{ij} = 1$ that indicates the relationship between node i and node j, otherwise $A_{ij} = 0$. $X \in \mathbb{R}^{n \times d}$ is the node attribute matrix, where each node v_i is associated with a d-dimensional vector x_i. Graph clustering aims to partition the n nodes into k clusters $\{C_1, C_2, \cdots, C_k\}$. The goal of clustering is maximizing inter-class similarity and minimizing intra-class similarity.

Next, our backbone (i.e., SDCN) is briefly de-

scribed. Reconstruction-based models generally contain two modules, namely the AE module and the GCN module. We summarize the objective functions for these two modules as follows.

The Reconstruction Loss. The reconstruction loss measures the mean square error of raw data and the reconstructed data which is formulated as Eq. (1).

$$L_{res} = \frac{1}{2n} \left\| X - \widehat{X} \right\|_F^2 \tag{1}$$

where $\widehat{X} = H^{(L)}$ is the reconstructed data. $H^{(L)} = f_{ae}(X)$ is the output of the AE module.

The Alignment Loss. The alignment loss aims to utilize the KL-divergence to measure the difference between different data distributions. The alignment loss includes clustering loss L_{clu} and graph neural network loss L_{gnn}. L_{clu} and L_{gnn} are formulated as follows:

$$L_{clu} = KL\left(P||Q\right), L_{gnn} = KL\left(P||Z\right) \tag{2}$$

where $Q = [q_{ij}]$ is the clustering result distribution, $P = [p_{ij}]$ is the auxiliary target distribution and $Z = U\left(f_{ae}(X), f_{gcn}(A, X)\right)$ is the probability distribution output by the backbone. $f_{gcn}(\cdot)$ is output of the GCN module. $U(\cdot)$ is a fusion function in the backbone, which is utilized to integrate the node representations obtained by GCN module and AE module. In addition, q_{ij} is the probability of sample i belongs to cluster j.

$$q_{ij} = \frac{\left(1 + \left\|h_i - c_j\right\|^2/\tau\right)^{-(\tau+1)/2}}{\sum_{j'}\left(1 + \left\|h_i - c_{j'}\right\|^2/\tau\right)^{-(\tau+1)/2}} \tag{3}$$

where h_i is the i-th of $H^{(L)}$ and c_j is the cluster center that initialized by a pre-trained AE. τ is set to 1. p_{ij} is formulated as follows:

$$p_{ij} = \frac{q_{ij}^2/\sum_i q_{ij}}{\sum_{j'} q_{ij'}^2/\sum_i q_{ij'}} \tag{4}$$

where $0 < p_{ij} < 1$. Combining Eq. (1) and Eq. (2), the learning objective function $L_{backbone}$ of the backbone can be obtained.

$$L_{backbone} = \beta_1 L_{res} + \beta_2 L_{clu} + \beta_3 L_{gnn} \tag{5}$$

where β_1, β_2 and β_3 are trade-off parameters which determined by the corresponding papers.

4 The Proposed Method

4.1 Multi-Granularity Contrastive Learning

In this subsection, we construct the multi-granularity contrastive learning modules and develop an adaptive feature fusion strategy to select unbiased positive and negative sample pairs.

As mentioned in section 3, reconstruction-based methods are selected as backbones. Meanwhile, InfoNCE, a widely used contrastive learning loss function, is adopted. The reason why we adopt InfoNCE will be discussed in subsection 4.3. Next, we will describe contrastive learning modules from graph-level and node-level, respectively.

4.1.1 Node-Level Module

The node-level module is designed to distinguish semantically similar (positive) and dissimilar (negative) node samples in the fine-grained node representations. In the node-level module, an adaptive feature fusion strategy is proposed to select positive and negative sample pairs, which contributes to alleviating the sampling bias.

Adaptive Feature Fusion Sampling. In this strategy, the graph structure information is regarded as prior knowledge, which can be dynamically integrated with node attribute features. According to graph structure information and node attribute feature, we defined two matrices, which are Structure Similarity Matrix $M_{SS} \in \mathbb{R}^{n \times n}$ and Feature Similarity Matrix $M_{FS} \in \mathbb{R}^{n \times n}$.

Specifically, M_{SS}^{ij} is defined as:

$$M_{SS}^{ij} = \frac{\left\|N\left(v_i\right) \cap N\left(v_j\right)\right\|}{\left\|N\left(v_i\right) \cup N\left(v_j\right)\right\|} \tag{6}$$

where $M_{SS}^{ij} \subset [0, 1]$, $N\left(v_i\right)$ is the neighbors of node i. In practice, Eq. (6) follows a simple assumption that node j is the 1-hop neighbor of node i, node r is the 1-hop neighbor of node j and the 2-hop neighbor of node i. If node j and node i do not belong to the same class, then node r and node i may not belong to the same class. Based on the above intuitive and strongly constrained assumption, $N\left(v_i\right)$ only considers 1-hop neighbors. Then, M_{FS} is calculated as:

$$M_{FS} = Z \cdot Z^T \tag{7}$$

where M_{FS}^{ij} measures the node feature similarity of node i and node j. Finally, we normalize M_{SS}, M_{FS} and weight them dynamically to obtain the Similarity Discrimination Matrix M_{SD}.

$$M_{SD} = \alpha M_{SS} + (1 - \alpha) M_{FS} \tag{8}$$

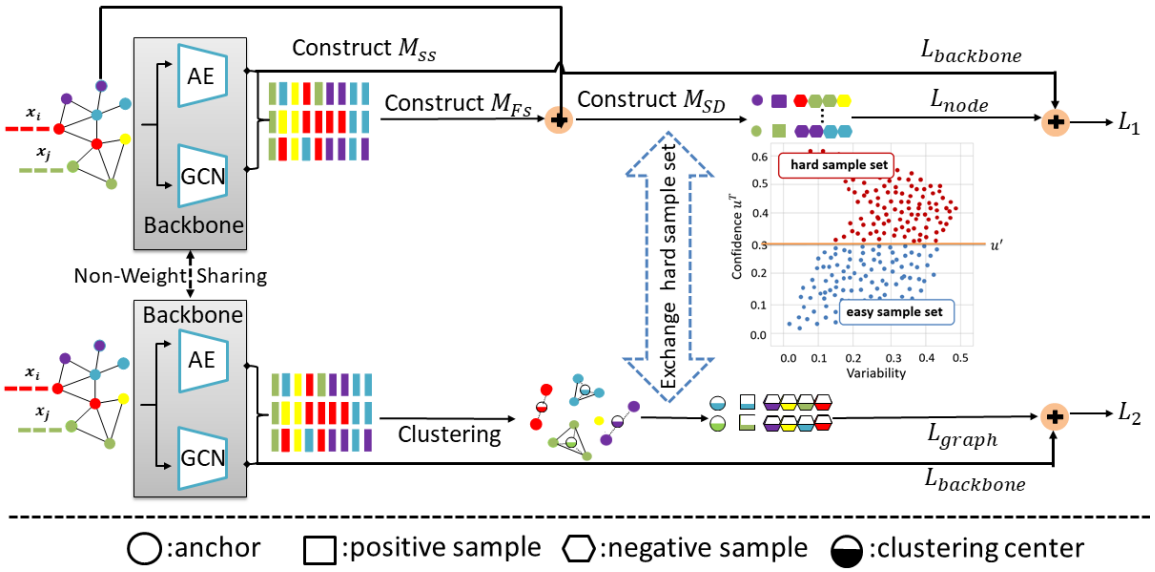


Figure 1: The framework of our proposed **DCL-MGI**.

where α is designed as an adaptive trade-off parameter, which is calculated by Eq. (9).

$$\alpha = \frac{1}{2}\left[KL\left(P||\frac{P+Z}{2}\right) + KL\left(Z||\frac{P+Z}{2}\right)\right] \quad (9)$$

Note that $\alpha = JS\left(P||Z\right)$ is the JS-divergence that measures the similarity between the P and Z distribution. In the early epoch, the poor model performance leads to the high dissimilarity between P and Z. Hence, M_{SD} tends to utilize the explicit graph structure information in the early epoch and focuses on the node features in the later epoch. In addition, since M_{SS} is fixed, even if the model suffers from over-fitting, M_{SD} still considers reliable graph structure information rather than relying entirely on incorrect node representations.

Hence, for v_i, the positive and negative samples can be selected based on M^i_{SD}. Specifically, one positive sample $\{v_i^p\}$ and $N-1$ negative samples $\left\{v_i^{n1}, \cdots v_i^{n(N-1)}\right\}$ are selected for each v_i. Then, the contrastive learning function for node-level module is calculated by Eq. (10).

$$L_{node} = \sum_{i=1}^{n} -\log \frac{f\left(v_i, v_i^p\right)}{f\left(v_i, v_i^p\right) + \sum_{j=1}^{N-1} f\left(v_i, v_i^{nj}\right)} \quad (10)$$

Note that $f(v_i, v_j) = \exp(\cos(g(v_i), g(v_j))/\tau)$. And $g\left(v_i\right)$ is the representation of node i generated by the backbone, which is equivalent to z_i in Z. τ is a temperature hyper-parameter that set to 1 for all experiments.

Based on the above discussion, we integrate the loss function of the backbone and the contrastive learning objective. Thus, the overall learning objective for the node-level module is formalized as:

$$L_1 = L_{backbone} + \lambda_1 L_{node} \quad (11)$$

where λ_1 is a trade-off parameter.

4.1.2 Graph-Level Module

Unlike the node-level module, the graph-level module focuses on the overall distribution of each class and aims to expand the inter-class dissimilarity. Specifically, we utilize the clustering center c_i^t to represent the distribution of i-th class at t-th epoch.

$$c_i^t = \frac{1}{|C_i^t|} \sum_{j \in C_i^t} z_j^t \quad (12)$$

where z_j^t is the representation of node j at t-th epoch and C_i^t is the node set of i-th class at the t epoch, respectively.

For c_i^t, the other clustering centers are selected as the negative samples $\left\{c_j^t | j \neq i, j \in [1, k]\right\}$. Meanwhile, c_i^{t-1} is selected as the positive sample for c_i^t. Hence, for each c_i^t, we construct one positive sample and $k-1$ negative samples, where k is determined by the specific downstream task. In addition, c_i^0 is initialized by performing K-means on a pre-trained AE output. Similarly, the contrastive learning function for graph-level module is calculated by Eq. (13)..

$$L_{graph} = \sum_{i=1}^{k} -\log \frac{f\left(c_i^t, c_i^{t-1}\right)}{f\left(c_i^t, c_i^{t-1}\right) + \sum_{j=1}^{k-1} f\left(c_i^t, c_j^t\right)} \quad (13)$$

Hence, the final learning objective for the graph-level module is formulated as:

$$L_2 = L_{backbone} + \lambda_2 L_{graph} \tag{14}$$

where λ_2 is a trade-off parameter.

4.2 Training Dynamics for Data Interaction

In this section, we study the integration method for 'communicate' the two peer contrastive learning modules as mentioned in subsection 4.1. Our motivation is that multi-granularity contrastive learning modules can learn the node representation from different views. In this way, the local and global information of the graph can be fused by interacting with the above two modules.

In general, samples that are frequently classified in the same class are easy to identify than those that vacillate. Hence, we regard the indistinguishable samples as the hard samples. To identify the hard samples, we bring in a statistical method arising from the behavior of the training procedure, which is named "training dynamics" (Swayamdipta et al., 2020). Unlike the method proposed by Swayamdipta, our method is designed for unsupervised scenarios.

Specifically, the temporal entropy information of node i is calculated, across T epochs. For node i, we first utilize the information entropy to measure the uncertainty at t-th epoch.

$$u_i^t = -\sum_{j=1}^{k} p_{\theta^t}\left(y_j|v_i\right) \log p_{\theta^t}\left(y_j|v_i\right) \tag{15}$$

where $p_{\theta^t}\left(y_j|v_i\right)$ indicates the probability distribution of the model output with parameters θ^t at the t-th epoch. The node with low uncertainty is easily distinguished. Then, we collect the historical information of u_i^t up to the T-th epoch to obtain u_i^T, where $u_i^T = \sum_{j=1}^{T} u_i^j$. Further, we set the threshold $u^{'}$ for u^T and divide the whole training dataset into hard sample set $\left\{v_i|\forall i \in [1,n], u_i^T > u'\right\}$ and easy sample set $\left\{v_i|\forall i \in [1,n], u_i^T \leq u'\right\}$.

Based on u_i^T and $u^{'}$, our proposed framework summarizes the training process into two stages as follows:

Independent learning stage. Two independent modules are trained separately. They share the same input data and train until the end of et epoch.

Information interaction stage. The hard sample set obtained by each module is exchanged to another. Keep exchanging the hard sample set until the end of the complete training. This stage is inspired by the active learning and co-teaching that realizes multi-granularity feature interaction.

Finally, the whole framework is illustrated in Figure 1 and summarized in Algorithm 1, respectively.

Algorithm 1 Training process of DCL-MGI

Input: Graph G, Maximum iterations $MaxIter$, Negative sample number N, Threshold $u^{'}$ and et
Output: The clustering result

```
Initialize node-level and graph-level modules.
for t = 0, 1, ..., MaxIter do
    if t <= et then
        Select positive and negative samples by Eq. (8).
        Calculate L_1 and L_2, respectively.
        Calculate u_i^t by Eq. (15).
        Update multi-granularity modules, separately.
    else
        Gather the historical information of u_i^t to get u_i^T.
        Divide the hard sample set based on u_i^T and u'.
        Update multi-granularity modules by interacting hard sample sets.
    end if
end for
Obtain the clustering results based on Z.
```

4.3 Why InfoNCE is Suitable for Clustering

In this section, we will briefly analyze the reason that InfoNCE can handle objective mismatch for clustering.

Given positive sample v_i^p and negative samples set $\left\{v_i^{nj}|j \subset [1, N-1]\right\}$ for node v_i. InfoNCE amis to minmize L_{cl}. The form of L_{cl} is shown as Eq. (10). Considering that minimizing L_{cl} is equivalent to maximizing $-L_{cl}$. Hence, we transform the goal as shown in Eq. (16)

$$\max -L_{cl} = \max \sum_{i=1}^{n} \log \frac{f\left(v_i, v_i^p\right)}{f\left(v_i, v_i^p\right) + \sum_{j=1}^{N-1} f\left(v_i, v_i^{nj}\right)} \tag{16}$$

Note that $\max \sum_{i=1}^{n} \log f\left(x\right) \Leftrightarrow \sum_{i=1}^{n} \max \log f\left(x\right)$. Hence, we can further simplify Eq. (16). Due to $\max \sum_{i=1}^{n} \log \frac{f\left(v_i, v_i^p\right)}{f\left(v_i, v_i^p\right) + \sum_{j=1}^{N-1} f\left(v_i, v_i^{nj}\right)}$ is equivalent to $\sum_{i=1}^{n} \max \log \frac{f\left(v_i, v_i^p\right)}{f\left(v_i, v_i^p\right) + \sum_{j=1}^{N-1} f\left(v_i, v_i^{nj}\right)}$ and it can

Dataset	# Type	# Samples	# Classes	# Dimension
ACM	Graph	3025	3	1870
Citeseer	Graph	3327	6	3703
DBLP	Graph	4057	4	334
USPS	Image	9298	10	256
HHAR	Record	10299	6	561
Reuters	Text	10000	4	2000

Table 1: The statistics of the benchmark datasets.

further be simplified to $\sum_{i=1}^{n} \max \log \frac{1}{1+\varphi}$, where $\varphi = \frac{\sum_{j=1}^{N-1} f\left(v_i, v_i^{nj}\right)}{f\left(v_i, v_i^{p}\right)}$.

Since $\log(\cdot)$ is a monotonically increasing function, maximizing $-L_{cl}$ approximates minimizing φ. By further simplification, the following approximate equation can be obtained, that is, $\min \frac{\sum_{j=1}^{N-1} f\left(v_i, v_i^{nj}\right)}{f\left(v_i, v_i^{p}\right)} \propto \frac{\min \sum_{j=1}^{N-1} f\left(v_i, v_i^{nj}\right)}{\max f\left(v_i, v_i^{p}\right)}$. Considering that if positive samples are selected from the same class of v_i, while negative samples are selected from the other $k-1$ classes. In that case, minimizing the L_{cl} is equivalent to the ratio of minimizing intra-class similarity and maximizing inter-class similarity, which is consistent with the objective of clustering.

Based on the above discussion, it is evident that introducing the InfoNCE loss function into the graph clustering task is suitable. Note that an important precondition is to construct the correct positive and negative sample pairs for each node. This precondition urges us to design the debiased contrastive sample selection strategy as mentioned in subsection 4.1.

4.4 Complexity Analysis

Time Complexity. In our proposed framework, the additional computational cost mainly comes from calculating M_{SD}, L_{node} and L_{graph}. For M_{SD}, the computational complexity is $O\left(n^2\right)$, that used to count 1-hop neighbors and matrix multiplication. Some graph traversal method (i.e., breadth first search) are adopted to construct M_{SS}. If multi-hop neighbors are considered, the time complexity will be further increased. Hence, we focus only on 1-hop neighbors. The computational complexity for L_{node} and L_{graph} are $O\left(nN\right)$ and $O\left(nk\right)$, where N and k are constants.

Space Complexity. In our proposed framework, the main space overhead comes from storing M_{SD}. If we store it naturally, then the space complexity is $o\left(n^2\right)$.

5 Experiments

5.1 Experiment Settings

Datasets. We evaluate the effectiveness of DCL-MGI framework on six benchmark datasets. Specifically, we adopt three classical graph datasets, including ACM, Citeseer, and DBLP. In addition, we also adopt three non-graph datasets, i.e, handwritten digit image dataset USPS (Hull, 1994), sensor record dataset HHAR (Stisen et al., 2015) and text news dataset Reuters (Lewis et al., 2004). For the above datasets, we follow the settings in (Bo et al., 2020). The statistics of benchmark datasets are shown in Table 1.

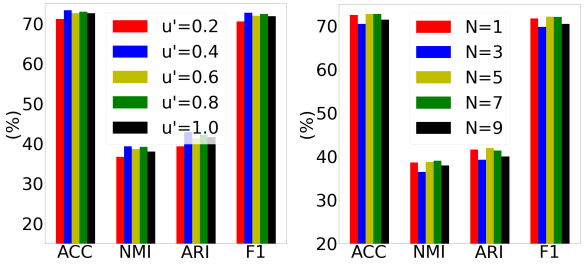
Baselines. We consider representative and state-of-the-art methods, including **RwSL** (Li et al., 2022), **DFCN** (Tu et al., 2021), **AGCN** (Peng et al., 2021), **SSGC** (Zhu and Koniusz, 2020), **SDCN** (Bo et al., 2020), **MVGRL** (Hassani and Khasahmadi, 2020), **AGRA** (Pan et al., 2019), **DAECG** (Wang et al., 2019), **VGAE** (Kipf and Welling, 2016). Note that DFCN, AGCN and SDCN are used as backbones. The combination of DCL-MGIand SDCN is denoted as DCL-MGI$_{\text{SDCN}}$ and DCL-MGI$_{\text{SD}}$, where DFCN and AGCN are similarly represented.

Evaluation Metrics. The evaluation metrics Accuracy (ACC), Normalized Mutual Information (NMI), Average Rand Index (ARI) and macro F1-score (F1) are adopted.

Parameters Setting. For backbones, we follow the same network structure and hyper-parameter settings with the corresponding paper. The learning rate is set to 0.001 for USPS, HHAR, ACM, and DBLP and 0.0001 for Reuters, Citeseer. The values of the hyper-parameters λ_1 and λ_2 are recorded in the appendix. For DCL-MGI$_{\text{SDCN}}$ and DCL-MGI$_{\text{AGCN}}$, the number of negative samples N is set to 5, the threshold of u' is set to 0.4, and the $MaxIter$ is set to 200. For DCL-MGI$_{\text{SDCN}}$ and DCL-MGI$_{\text{DFCN}}$, the number of negative samples N is set to 9, the threshold of u' is set to 0.2, and the $MaxIter$ is set to 300. The number of epochs in the first stage et is set to 120 for all experiments. For SDCN and AGCN, we report the highest evaluation scores among all variants. For AGCN, we record experimental results by running the official code. For other comparisons, we directly cite the results from the original papers (Peng et al., 2021; Bo et al., 2020; Liu et al., 2021). For each experiment, we run 10 times and report the average values to prevent extreme cases.

Table 2: Clustering performance (%) on the benchmark datasets (mean±std). The best results are shown in bold. ↑ records the improvement over the backbones.

Dataset	Metric	VGAE	DAEGC	ARGA	MVGRL	SSGC	RwSL	SDCN	DCL-MGI$_{SD}$	↑	AGCN	DCL-MGI$_{AG}$	↑	DFCN	DCL-MGI$_{DF}$	↑
DBLP	ACC	58.6±0.1	62.1±0.5	61.6±1.0	42.7±1.0	68.7±2.0	68.3±0.5	68.1±1.8	72.8±1.2	4.7	71.6±1.0	73.1±0.7	1.5	76.0±0.8	**76.7±0.7**	0.7
	NMI	26.9±0.1	32.5±0.5	26.8±1.0	15.4±0.6	33.9±2.1	34.4±0.4	39.5±1.3	39.8±0.7	0.3	37.6±1.3	39.2±0.7	1.6	43.7±1.0	**44.5±0.1**	0.8
	ARI	17.9±0.1	21.0±0.5	22.7±0.3	8.2±0.2	37.3±3.1	34.5±0.8	39.2±2.0	41.7±0.9	2.5	40.5±1.2	42.0±1.0	1.5	47.0±1.5	**48.0±0.2**	1.0
	F1	58.7±0.1	61.8±0.7	61.8±0.9	40.5±1.5	65.9±2.2	68.2±0.5	67.7±1.5	71.9±1.4	4.2	71.2±1.0	72.8±0.6	1.6	75.7±0.8	**76.5±0.1**	0.8
CiteSeer	ACC	61.0±0.4	64.5±1.4	56.9±0.7	68.7±0.4	67.9±0.3	70.2±0.1	66.0±0.3	69.5±0.3	3.5	68.7±0.3	68.9±0.1	0.2	69.5±0.2	**70.3±0.1**	0.8
	NMI	32.7±0.3	36.4±0.9	34.5±0.8	43.7±0.4	41.9 ±0.2	44.3 ±0.2	38.7±0.3	41.8±1.6	3.1	41.5±0.2	41.7±0.1	0.2	43.9±0.2	**44.6±0.1**	0.7
	ARI	33.1±0.5	37.8±1.2	33.4±1.5	44.3±0.7	43.0±0.3	46.1±0.2	40.2±0.4	44.5±1.7	4.3	43.5±0.3	43.9±0.1	0.4	45.5±0.3	**46.6±0.1**	1.1
	F1	57.7±0.5	62.2±1.3	54.8±0.8	63.7±0.4	63.6±0.2	**66.1±0.1**	63.6±0.2	63.8±0.9	0.2	62.4±0.2	62.5±0.2	0.1	64.3±0.2	65.0±0.2	0.7
ACM	ACC	84.1±0.2	86.9±2.8	86.1±1.2	86.7±0.8	84.4±0.3	90.7±0.1	90.5±0.2	90.8±0.2	0.3	90.0±0.5	90.3±0.2	0.3	90.9±0.2	**91.3±0.2**	0.4
	NMI	53.2±0.5	56.2±4.2	55.7±1.4	60.9±1.4	56.2±0.5	69.1±0.1	68.3±0.3	68.7±0.6	0.4	66.8±1.2	68.1±0.3	1.3	69.4±0.4	**71.0±0.2**	1.6
	ARI	57.7±0.7	59.4±3.9	62.9±2.1	65.1±1.8	60.2±0.6	74.5±0.1	73.9±0.4	74.6±0.6	0.7	72.5±1.2	73.6±0.4	1.1	74.9±0.4	**76.2±0.2**	1.3
	F1	84.2±0.2	87.1±2.8	86.1±1.2	86.9±0.7	84.4±0.3	90.7±0.1	90.4±0.2	90.8±0.2	0.4	90.0±0.5	90.3±0.2	0.3	90.8±0.2	**91.3±0.2**	0.5
USPS	ACC	56.2±0.7	73.6±0.4	66.8±0.7	-	-	-	78.1±0.2	80.6±0.7	2.5	80.2±0.4	**81.0±0.1**	0.8	79.5±0.2	79.6±0.1	0.1
	NMI	51.1±0.4	71.1±0.2	61.6±0.3	-	-	-	79.5±0.3	79.8±0.4	0.3	79.1±0.3	79.5±0.3	0.4	82.8±0.3	**83.3±0.1**	0.5
	ARI	41.0±0.6	63.3±0.3	51.1±0.6	-	-	-	71.8±0.2	73.5± 0.5	1.7	72.6±0.5	73.7±0.2	1.1	75.3±0.2	**75.7±0.2**	0.4
	F1	53.6±1.1	72.5±0.5	66.1±1.2	-	-	-	77.0±0.2	78.1±0.2	1.1	77.0±0.3	77.5±0.4	0.5	78.3±0.2	**78.5±0.1**	0.2
HHAR	ACC	71.3±0.4	76.5±2.2	63.3±0.8	-	-	-	84.3±0.2	87.5±0.9	3.2	88.0±0.1	**88.4±0.4**	0.4	87.1±0.1	87.2±0.1	0.1
	NMI	63.0±0.4	69.1±2.3	57.1±1.4	-	-	-	79.9±0.1	81.2±0.4	1.3	**82.6±0.7**	82.1±0.3	-0.5	82.2±0.1	82.4±0.1	0.2
	ARI	51.5±0.7	60.4±2.2	44.7±1.0	-	-	-	72.8±0.1	76.2±1.4	3.4	77.0±0.4	**77.5±0.5**	0.5	76.4±0.1	76.5±0.1	0.1
	F1	71.6±0.3	76.9±2.2	61.1±0.9	-	-	-	82.6±0.1	86.5 ±1.2	3.9	87.9±0.5	**88.2±0.5**	0.3	87.3±0.1	87.5±0.1	0.2
Reuters	ACC	60.9±0.2	65.5±0.1	56.2±0.2	-	-	-	79.3±0.1	80.7±0.6	1.4	80.8±0.4	**81.2±0.1**	0.4	77.7±0.2	78.1±0.1	0.4
	NMI	25.5±0.2	30.6±0.3	28.7±0.3	-	-	-	56.9±0.3	58.8±0.5	1.9	59.6±0.3	60.1±0.2	0.5	59.9±0.4	**60.7±0.1**	0.8
	ARI	26.2±0.4	31.1±0.2	24.5±0.4	-	-	-	59.6±0.3	62.5±1.1	2.9	61.2±0.9	**62.8±0.7**	1.6	59.8±0.4	60.4±0.1	0.6
	F1	57.1±0.2	61.8±0.1	51.1±0.2	-	-	-	66.2±0.2	66.8±0.4	0.6	65.6±0.2	66.7±0.7	1.1	69.6±0.1	**69.8±0.0**	0.2



(a) u' (b) N

Figure 2: The parametric sensitivity analysis of DCL-MGI$_{SDCN}$ on DBLP.

5.2 Graph Clustering Results

Table 2 reports the clustering results on six benchmark datasets. From Table 2, we can see that DCL-MGI is easily combined with different backbones and further improves their original performance. For example, for the non-graph dataset HHAR, DCL-MGI improves upon the original SDCN by 3.2%, 2.3%, 3.4%, 3.9% in terms of ACC, NMI, ARI, and F1, respectively. For graph dataset DBLP, DCL-MGI improves upon the original AGCN by 1.5%, 1.6%, 1.5%, 1.6% on ACC, NMI, ARI, and F1, respectively. Meanwhile, DCL-MGI$_{SDCN}$ also improves 4.7% on ACC and 4.2% on F1 for DBLP. These significant improvements can be attributed to two keys: (1). The objective of DCL-MGI is designed for clustering and the selected contrastive sample pairs are unbiased. (2). DCL-MGI integrates graph-level and node-level graph information by interacting with hard samples. In section 5.5, the validity of the interaction hard sample is further demonstrated.

5.3 Parameter Sensitivity Analysis

As depicted in Figure 2, we consider the threshold of uncertainty u' and the number of negative samples N, where $u^{'} = \{0.2, 0.4, 0.6, 0.8, 1.0\}$ and $N = \{1, 3, 5, 7, 9\}$. Meanwhile, we adopt DCL-MGI$_{SDCN}$ and conduct experiments on DBLP. From Figure 2(a), we see that DCL-MGI$_{SDCN}$ reaches the best results when $u^{'}$ is 0.4. From Figure 2(b), it can be seen that ACC and NMI obtain the best result when N is 7 and ARI obtain the best result when N is 5. On the whole, DCL-MGI$_{SDCN}$ is insensitive to the above parameters. In addition, we further explore the parameter sensitivity of λ_1, λ_2 and et. The results are recorded in the appendix.

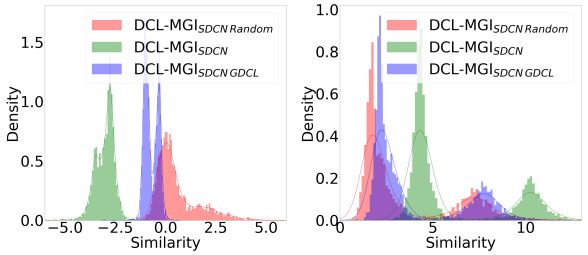
5.4 Ablation Study

We conduct ablation studies for DCL-MGI$_{SDCN}$ variants and evaluate on DBLP. The results are recorded in Table 3.

Contrastive Sample Selection Strategy. DCL-MGI$_{SDCN\ Random}$ adopts the random sampling which used in (Hassani and Khasahmadi, 2020) and DCL-MGI$_{SDCN\ GDCL}$ adopts the node clustering sampling which proposed in (Zhao et al., 2021). For our proposed adaptive feature fusion strategy, DCL-MGI$_{SDCN\ Topology}$ utilizes only graph structure information M_{SS} and DCL-MGI$_{SDCN\ Feature}$ utilizes only node attribute feature M_{FS}. The results show that our proposed contrastive sample selection strategy contributes to achieve optimal model performance. DCL-MGI$_{SDCN\ Topology}$ achieves the lowest model performance beacuse it only utilizes 1-hop neighbors information. However, DCL-MGI$_{SDCN\ Topology}$ still achieves better

Table 3: Clustering performance (%) for the different DCL-MGI$_{\text{SDCN}}$ variants (mean±std).

Variants	ACC	NMI	AIR	F1
DCL-MGI$_{\text{SDCN Random}}$	71.1±1.0	37.0±0.9	39.8±1.3	70.7±0.9
DCL-MGI$_{\text{SDCN GDCL}}$	71.7±0.9	37.7±1.4	40.4±1.7	70.9±0.5
DCL-MGI$_{\text{SDCN Topology}}$	69.9±1.7	35.3±2.1	37.8±2.6	68.8±1.8
DCL-MGI$_{\text{SDCN Feature}}$	72.3±1.1	38.3±1.5	41.5±1.7	71.1±1.5
DCL-MGI$_{\text{SDCN Graph}}$	70.1±0.8	35.7±1.2	38.9±2.4	69.2±1.8
DCL-MGI$_{\text{SDCN Node}}$	70.8±1.8	36.5±2.2	38.9±1.9	70.1±1.9
DCL-MGI$_{\text{SDCN Triplet}}$	72.5±1.4	38.5±1.7	41.9±1.9	71.5±1.4
DCL-MGI$_{\text{SDCN}}$	72.8±1.2	39.8±0.7	41.7±0.9	71.9±1.4



(a) Negative Samples (b) Positive Samples

Figure 3: Similarity distribution of contrastive learning sample pairs on DBLP.

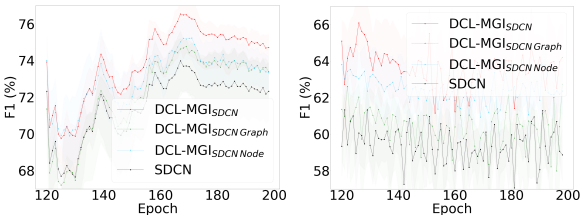
performance than backbone.

Multi-Granularity Contrastive Modules. DCL-MGI$_{\text{SDCN graph}}$ removes the node-level module and DCL-MGI$_{\text{SDCN node}}$ removes the graph-level module, which limits them to learn node representation from single view. The results indicate that all collaborative training methods except DCL-MGI$_{\text{SDCN Topology}}$ achieve better performance than DCL-MGI$_{\text{SDCN Graph}}$ and SDCN$_{\text{w/o Node}}$. This phenomenon indicates that interacting hard samples for multi-granularity feature interaction is beneficial to learn more distinguished node representations.

Contrastive Learning Objective Function. DCL-MGI$_{\text{SDCN Triplet}}$ use the Triplet (Chopra et al., 2005) loss instead of InfoNCE. The results indicate that our framework does not rely on a specific objective function and is well suited for different learning objectives.

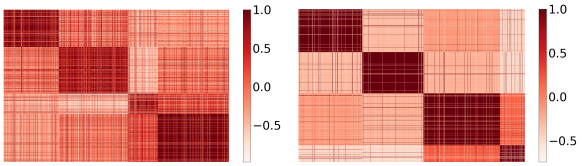
5.5 Qualitative Study

Similarity Distribution. To further explore the data distribution on contrastive sample pairs. We calculate the similarity of negative and positive samples to anchor by the inner product. The results are shown in Figure 3. Figure 3(a) depicts that the negative samples selected by adaptive feature fusion are furthest from the anchor. Similarly, Figure 3(b) shows that the positive samples selected by



(a) Easy Samples (b) Hard Samples

Figure 4: The F1 metric across the information interaction stage on DBLP.



(a) SDCN (b) DCL-MGI$_{\text{SDCN}}$

Figure 5: The heat maps of feature similarity on DBLP.

our proposed strategy have the highest similarity to the anchor. These results further demonstrate that the adaptive feature fusion strategy can effectively alleviate the sample bias.

Hard Sample Interaction Strategy. We study the effectiveness of the hard sample interaction strategy. We conduct experiments on DBLP and the results are shown in Figure 4. From Figure 4(a), our proposed model achieves the best performance on the easy dataset, and the model performance is further improved across the model training. Similarly, DCL-MGI$_{\text{SDCN}}$ still obtains the best performance on the hard dataset. This further confirms the effectiveness of multi-granularity feature interaction.

Node Feature Similarity. We extract the node features and visualize the similarity matrices calculated by the cosine similarity. Figure 5 shows our proposed method further improves the discrimination of node features. The results demonstrate that our proposed framework can alleviate over-fitting.

6 Conclusion

In this paper, we propose a novel and flexible self-supervised deep graph clustering framework DCL-MGIwith unbiased sampling and multi-granularity feature interaction mechanisms. It consists of two parallel contrastive learning modules and utilizes an adaptive feature fusion strategy for selecting unbiased contrastive sample pairs. Further, a temporal entropy-based metric is proposed for effective interaction between multi-granularity features. Extensive experiments prove the effectiveness of our framework.

7 Limitations

In this paper, two individual contrastive learning modules require more computation time and memory space. Tacking DFCN as an example, DCL-MGI_{DFCN} runs 210.23 seconds on the Citeseer dataset, while DFCN runs 56.49 seconds. DCL-MGI_{DFCN} runs 210.23 seconds on the Citeseer dataset, while DFCN runs 56.49 seconds. DFCN stores 1.91M model parameters and DCL-MGI_{DF} stores 3.82M model parameters. In the future, we will utilize parameter sharing to reduce the number of training parameters.

References

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

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *ICML*, pages 1597–1607.

Sumit Chopra, Raia Hadsell, and Yann LeCun. 2005. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, pages 539–546.

Ching-Yao Chuang, Joshua Robinson, Lin Yen-Chen, Antonio Torralba, and Stefanie Jegelka. 2020. Debiased contrastive learning. In *NIPS*, pages 8765–8775.

Kan Guo, Yongli Hu, Yanfeng Sun, Sean Qian, Junbin Gao, and Baocai Yin. 2021. Hierarchical graph convolution networks for traffic forecasting. In *AAAI*, pages 151–159.

Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *AISTATS*, pages 297–304.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *CVPR*, pages 1735–1742.

Kaveh Hassani and Amir Hosein Khasahmadi. 2020. Contrastive multi-view representation learning on graphs. In *ICML*, pages 4116–4126.

Qian Huang, Horace He, Abhay Singh, Ser-Nam Lim, and Austin Benson. 2020. Combining label propagation and simple models out-performs graph neural networks. In *ICLR*.

Jonathan J. Hull. 1994. A database for handwritten text recognition research. *IEEE Transactions on pattern analysis and machine intelligence*, 16(5):550–554.

Yuheng Jia, Junhui Hou, and Sam Kwong. 2020. Constrained clustering with dissimilarity propagation-guided graph-laplacian pca. *IEEE Transactions on Neural Networks and Learning Systems*, 32(9):3985–3997.

Thomas N Kipf and Max Welling. 2016. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*.

Lingpeng Kong, Cyprien de Masson d'Autume, Lei Yu, Wang Ling, Zihang Dai, and Dani Yogatama. 2019. A mutual information maximization perspective of language representation learning. In *ICLR*.

David D Lewis, Yiming Yang, Tony Russell-Rose, and Fan Li. 2004. Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 5(4):361–397.

Xiang Li, Dong Li, Ruoming Jin, Gagan Agrawal, and Rajiv Ramnath. 2022. Scalable deep graph clustering with random-walk based self-supervised learning. In *WWW*.

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

Yue Liu, Wenxuan Tu, Sihang Zhou, Xinwang Liu, Linxuan Song, Xihong Yang, and En Zhu. 2021. Deep graph clustering via dual correlation reduction. *arXiv preprint arXiv:2112.14772*.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Erlin Pan and Zhao Kang. 2021. Multi-view contrastive graph clustering. In *NIPS*.

Shirui Pan, Ruiqi Hu, Sai-fu Fung, Guodong Long, Jing Jiang, and Chengqi Zhang. 2019. Learning graph embedding with adversarial training methods. *IEEE transactions on cybernetics*, 50(6):2475–2487.

Zhihao Peng, Hui Liu, Yuheng Jia, and Junhui Hou. 2021. Attention-driven graph clustering network. In *MM*, pages 935–943.

Allan Stisen, Henrik Blunck, Sourav Bhattacharya, Thor Siiger Prentow, Mikkel Baun Kjærgaard, Anind Dey, Tobias Sonne, and Mads Møller Jensen. 2015. Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition. In *SenSys*, pages 127–140.

Siqi Sun, Zhe Gan, Yuwei Fang, Yu Cheng, Shuohang Wang, and Jingjing Liu. 2020. Contrastive distillation on intermediate representations for language model compression. In *EMNLP*, pages 498–508.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *EMNLP*, pages 9275–9293.

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

Chun Wang, Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, and Chengqi Zhang. 2019. Attributed graph clustering: A deep attentional embedding approach. In *IJCAI*, pages 3670–3676.

Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *SIGIR*, pages 726–735.

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

Hao Zhu and Piotr Koniusz. 2020. Simple spectral graph convolution. In *ICLR*.

A Appendix

A.1 Experimental environment

We carried out the experiment on the window platform with Inter(R) Core(TM) i7-10700 CPU, RTX 3090 GPU, and 32G memory.

A.2 License

The backbones and the benchmark datasets can be used for academic research under the corresponding paper license.

A.3 Parameter Settings

We record the hyper-parameters λ_1 and λ_2 as shown in Table 4, Table 5 and Table 6.

Table 4: The parameter settings of DCL-MGI$_{\text{SDCN}}$.

Dataset	λ_1	λ_2
USPS	10	0.01
HHAR	1	0.01
Reuters	10	1000
ACM	1	0.01
DBLP	1000	10
Citeseer	100	0.1

Table 5: The parameter settings of DCL-MGI$_{\text{AGCN}}$.

Dataset	λ_1	λ_2
USPS	0.001	100
HHAR	1000	10
Reuters	0.1	0.001
ACM	100	1
DBLP	0.01	10
Citeseer	1	0.001

Table 6: The parameter settings of DCL-MGI$_{\text{FDCN}}$.

Dataset	λ_1	λ_2
USPS	0.001	100
HHAR	1000	10
Reuters	1000	1000
ACM	0.01	0.1
DBLP	0.1	0.1
Citeseer	0.1	1000

As described in Section 5.1, the other parameters N, $u^{'}$ and et are fixed for all experiments.

A.4 Parameter Sensitivity Analysis

We show The parametric sensitivity analysis for et in Figure 6. Further, we record the value of metrics for λ_1 and λ_2 in the range of $\{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$. The results are shown in Figure 7. Meanwhile, a numerical statistical analysis of Figure 7 is carried out and the results are recorded in Table 7.

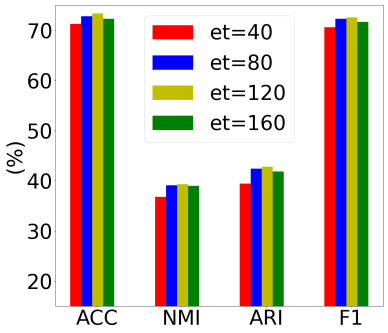
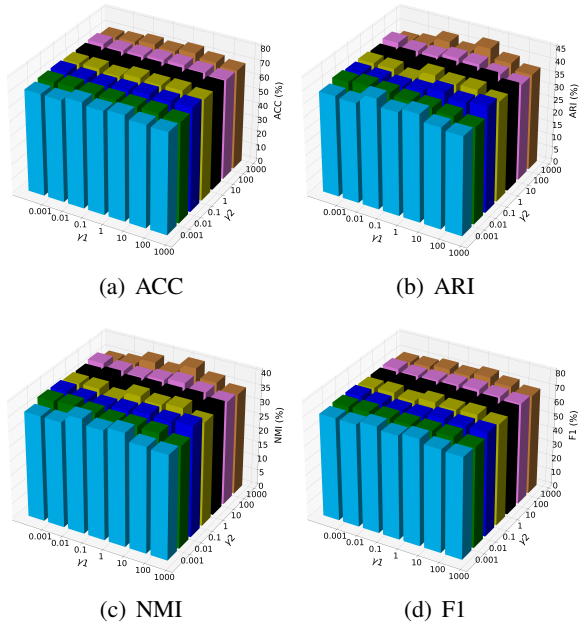


Figure 6: The parametric sensitivity analysis for et on DBLP.



(a) ACC (b) ARI

(c) NMI (d) F1

Figure 7: Parametric sensitivity analysis for λ_1 and λ_2 on DBLP.

Table 7: The numerical statistics of Figure 7

Metrics	Mean	Std	Max	Min
ACC	71.4	0.9	73.8	69.3
ARI	40.1	1.3	43.3	37.3
NMI	37.2	1.1	40.2	34.9
F1	71.4	1.1	73.2	68.9