Information Processing and Management 62 (2025) 104084

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

MHGC: Multi-scale hard sample mining for contrastive deep graph clustering☆

Tao Ren [a], Haodong Zhang [a], Yifan Wang [b,*], Wei Ju [c], Chengwu Liu [c], Fanchun Meng [a], Siyu Yi [d], Xiao Luo [e]

[a] Software College, Northeastern University, Shenyang, 110170, China
[b] School of Information Technology & Management, University of International Business and Economics, Beijing, 100029, China
[c] School of Computer Science, Peking University, Beijing, 100871, China
[d] School of Statistics and Data Science, Nankai University, Tianjin, 300071, China
[e] Department of Computer Science, University of California, Los Angeles, 90095, USA

ARTICLE INFO

Keywords:
Deep graph clustering
Graph contrastive learning
Multi-scale representation
Hard sample mining

ABSTRACT

Contrastive graph clustering holds significant importance for numerous real-world applications and yields encouraging performance. However, current efforts often overlook hierarchical high-order semantic information and treat all contrastive pairs equally during optimization. Consequently, the abundance of well sample pairs overwhelms the critical structural context learning process, limiting the accumulation of information and deteriorating the network's learning capability. To address this concern, a novel contrastive deep graph clustering method termed MHGC is proposed by conducting hard sample mining in contrastive learning with multi-granularity. Specifically, random walk with restart is utilized to sample subgraphs centered around anchor nodes. Then, an attribute encoder to learn node representations is designed to obtain subgraph embeddings. Subsequently, hard and easy sample pairs within high-confidence clusters is identified by applying a two-component beta mixture model to the clustering loss. Building upon this, a weight regulator is then elaborated to adaptively tune the weights of sample pairs and a multi-scale contrastive loss framework is proposed to leverage structural context information in a hierarchical contrastive manner. Comprehensive experiments conducted on six widely used datasets confirm the comparable performance of our MHGC relative to the state-of-the-art baselines, demonstrating an average increase of 1.54% in accuracy. Additionally, the ablation study further proves that our proposed multi-scale learning scheme and BMM-based hard mining strategy are effective approaches for the graph clustering task. The source code is available at https://github.com/sodarin/MHGC

1. Introduction

Graphs is an essential data structure for representing complex relationships between objects and have been extensively studied over the years. Especially graph learning algorithms, which utilize machine learning to derive significant features from graphs, have gained exceptional success in various tasks like node/graph classification (Duan et al., 2024; Ju et al., 2024a; Mao et al.,

☆ This work is partially supported by "Fundamental Research Funds for the Central Universities, China (N2217003)", and "the Fundamental Research Funds for the Central Universities" in UIBE (Grant No. 23QN02).

* Corresponding author.
E-mail address: yifanwang@uibe.edu.cn (Y. Wang).

https://doi.org/10.1016/j.ipm.2025.104084
Received 17 October 2024; Received in revised form 30 December 2024; Accepted 28 January 2025
Available online 13 February 2025

2023; Wang et al., 2024, 2023; Wu et al., 2022; Yuan et al., 2023), link prediction (Halliwell, 2022; Roy et al., 2021; Wang et al., 2022a, 2025) and anomaly detection (Duan et al., 2023; Liu, Ding et al., 2023). Within the diverse landscape of graph learning, one core and challenging task that has recently attracted considerable interest is graph clustering, serving as a highlighting prerequisites in multiple real-world applications, such as domain adaptation (Luo et al., 2024a, 2024b; Ma et al., 2019; Yuan et al., 2022), representation learning (Ju et al., 2023a; Shang et al., 2024; Zhang et al., 2024), and community detection (Hu et al., 2019; Wang et al., 2021), anomaly detection (Duan et al., 2023; Liu, Ding et al., 2023).

The primary concept of graph clustering is to divide the nodes of a graph into separate groups, ensuring that nodes within the same cluster exhibit higher resemblance to one another than to those in different clusters. Due to the robust representation capacity of deep learning, especially graph neural networks (GNNs) (Ju et al., 2024c; Kipf & Welling, 2017; Xu et al., 2019), numerous deep graph clustering methods have been continuously developed, generally falling into three main categories: reconstructive-based methods (Mrabah et al., 2022; Wang et al., 2019, 2017; Yi et al., 2023), adversarial-based methods (Gong et al., 2022; Pan et al., 2019; Tao et al., 2019) and contrastive learning methods (Cui et al., 2020; Hassani & Khasahmadi, 2020; Liu, Yang, Zhou, Liu, Wang, Liang, Tu, Li, Duan et al., 2023; Xia, Wang et al., 2022; Yang et al., 2023). Reconstructive-based and adversarial-based methods learn cluster-oriented node representations through mechanisms such as recovering graph information and generating and recognizing fake samples, respectively. Since most of these methods typically leverage the original topology structure as the input for the learning process, their performance is highly susceptible to the potential noisy or low-quality input graph, leading to sub-optimal and inconsistent results.

Recently, contrastive learning methods have shown significant promise in deep graph clustering. By augmenting the distance between negative sample pairs while concurrently reducing the distance between positive sample pairs, these methods can implicitly acquire supervisory information without necessitating human annotations, resulting in loss functions for graph clustering that are more coherent and discriminative. For example, SCGC (Liu, Yang, Zhou, Liu, Wang, Liang, Tu and Li, 2023) proposes a simple neighbor-oriented contrastive loss to maintain structural consistency across different views. CCGC (Yang et al., 2023) performs contrastive learning guided by the high-confidence clustering pseudo labels to improve the graph clustering result. HSAN (Liu, Yang, Zhou, Liu, Wang, Liang, Tu, Li, Duan et al., 2023) considers both the information implicit in the attribute and the structure of the graph, focusing on the hard sample mining to compute the similarity.

Although these recent efforts have achieved significant success, they suffer all or at least partially the following limitations. *(1) Fail to effectively capture high-order semantics.* Existing works mostly focus on the perspective of a single node scale contrast and have overlooked the potential for further exploration of subgraph information. We argue that nodes and their local neighbors exhibit stronger correlations, while distant nodes have minimal influence. Subgraphs consisting of regional neighborhoods are crucial for providing high-order structure contexts and have demonstrated benefits for graph-based machine learning tasks (Duan et al., 2023; Jiao et al., 2020). (2) *Fail to sufficiently capture hard boundary samples.* Most contrastive methods treat all samples as equally significant and overlook hard yet important boundary samples in the graph clustering task. This issue is particularly pronounced in multi-scale graph learning methods, which will lead to a model bias towards either global or highly localized representations. As a result, it is often insufficient to distinguish hard samples with only a single scale perspective.

Towards this end, we propose MHCG in this paper, a new **M**ulti-scale **H**ard sample mining framework for deep **G**raph **C**lustering, which considers the hardness correlation between anchor nodes and their regional subgraphs. Specifically, we consider the original graph and its edge modification version as two views of the graph. In each view, subgraphs are sampled through random walks, and a hierarchical multi-scale contrastive scheme is proposed to preserve multi-level information in the representation. Then, we employ a clustering layer to align the learned representation with the clustering results. By fitting a two-component (true-false) beta mixture model (BMM) on the clustering loss, we can distinguish high-confidence nodes. Consequently, samples within the same clusters that show low similarity, along with those from different clusters that display high similarity, are identified as potential hard samples. In our hierarchical contrastive scheme, we increase the weights of hard sample pairs while decreasing those of easier ones as the granularity becomes finer, which further enhances the discriminative capacity of samples across multiple scales for the graph clustering task. The contributions of this paper can be summarized as follows:

- *Conceptual:* We incorporate the subgraph information to offer high-order structural contexts for graph clustering and highlight the hard samples from a multi-scale perspective. The ablation study on the combinations of each level demonstrate the contributions of leveraging multi-scale structural contexts.
- *Methodological:* We propose a hierarchical multi-scale contrastive learning scheme with a graph clustering layer and utilize a BMM to estimate the high-confidence nodes, which further guides hard sample mining and, in turn, dynamically weights the learning scheme. The ablation study and the parameter analysis of τ both highlight a promising improvement of discriminative capability provided by the BMM-based mining strategy.
- *Experimental:* We perform comprehensive experiments on various public datasets to evaluate the performance of MHGC. Experimental results highlight the superiority and effectiveness of our proposed framework for the deep graph clustering task with an average increase of 1.54% in terms of accuracy.

2. Related work

2.1. Deep graph clustering

Deep graph clustering has emerged as a popular framework in recent years, with the goal of partitioning the nodes of a graph into disjoint clusters using neural networks (Wang et al., 2017; Xia et al., 2021; Zhang et al., 2020). From a learning paradigm perspective,

deep graph clustering roughly falls into three categories: (1) reconstructive-based methods; (2) adversarial-based methods; and (3) contrastive learning methods. Our model is primarily focused on the last contrastive learning methods, and we dedicates the next subsection to provide a detailed review of these methods.

Reconstructive-based approaches exploit intra-data information within the graph and learn representations by integrating both graph structure and the node attributes (Mrabah et al., 2022; Wang et al., 2019, 2017). For example, MGAE (Wang et al., 2017) utilizes a graph-based auto-encoder (GAE) to generate hidden representations of nodes. DAEGC (Wang et al., 2019) combines attention mechanism and auto-encoder to capture the topological information. More recently, R-GAE (Mrabah et al., 2022) proposes a novel scheme to alleviate the feature randomness and drift issues, enhancing the performance of existing GAE-based methods.

Adversarial-based methods aim to provide promising node representations by establishing a adversarial training scheme between the generator and the discriminator (Gong et al., 2022; Pan et al., 2019; Tao et al., 2019). For example, AGAE (Tao et al., 2019) incorporates adversarial mechanism into auto-encoder to improve the discriminative capability of samples. ARGA (Pan et al., 2019) employs adversarial training to ensure that latent features align with a prior distribution. AGC-DRR (Gong et al., 2022) introduces adversarial learning to adaptively learn the adjacency matrix for attributed graph clustering.

Although these methods have been proven to be effective, they depend heavily on the quality of the input topology structure for graph learning, since they are trained by reconstructing the objectives of adjacency matrix or feature matrix (Gong et al., 2022; Mrabah et al., 2022). Therefore, this reliance will force the model to remember high-frequency noises in features introduced from the original graph, leading to suboptimal performance (Cui et al., 2020; Wang et al., 2022b). In contrast, contrastive learning methods (Pan & Kang, 2021; Xia, Wang et al., 2022; Yang et al., 2023) focus on gathering supervisory information from the augmented view of the original graph in a contrastive manner, thereby alleviating the impact of a low-quality input graph.

2.2. Graph contrastive learning

Graph contrastive learning is an effective method that reduces the reliance on labeled data by exploring the intrinsic consistency between different views of a graph (Ju et al., 2024). Existing efforts in graph contrastive learning have primarily applied to fields such as drug discovery (Gu et al., 2023; Ju et al., 2023), recommender systems (Chen et al., 2023; Li et al., 2024; Mao et al., 2021), and out-of-distribution detection (Duan et al., 2023; Liu, Ao et al., 2023; Liu, Ding et al., 2023). Concretely, HSL-RG (Ju et al., 2023) builds relation graphs using graph kernels to learn transformation-invariant representations in a self-supervised manner. HGCL (Chen et al., 2023) leverages heterogeneous relational semantics to model user-item interactions using contrastive learning, enhancing the knowledge transfer across different views of the graph. GOOD-D (Liu, Ding et al., 2023) introduces a hierarchical contrastive learning on augmented graphs to capture semantic inconsistencies across multiple levels of granularity.

Recently, the application of contrastive learning on graph clustering has garnered considerable attention which enhances the discriminativeness of features by separating negative samples further apart while pulling positive samples closer (Cui et al., 2020; Hassani & Khasahmadi, 2020; Xia, Wang et al., 2022; Yang et al., 2023). Specifically, GDCL (Zhao et al., 2021) utilizes contrastive learning to align the negative and positive samples from different views. SCAGC (Xia, Wang et al., 2022) improves the quality of negative pairs by randomly sampling nodes from the distinct clusters. CGCC (Yang et al., 2023) leverages contrastive learning to learn representations by contrasting negative and positive samples derived from high-confidence clustering pseudo labels.

In spite of their success on extensive benchmarks, most of these works assume that all samples contribute equally, neglecting the influence of hard samples. As a result, they may suffer from a loss of discriminative capability. We suggest a unique BMM-based hard sample mining strategy to overcome this constraint by dynamically modifying the weights assigned to hard samples, thereby enhancing the reliability of the sample pairs and improving overall performance.

2.3. Graph hard sample mining

Hard negative mining, which serves as a selection strategy for negative and positive samples, is essential for achieving promising performance by emphasizing hard negatives. Previous approaches (Chuang et al., 2020; Kalantidis et al., 2020; Robinson et al., 2020) on images have demonstrated that hard negative mining is a promising approach to further boost the performance of contrastive learning. Motivated by their success, many efforts have tried to incorporate hard negative mining into deep graph clustering to tackle the challenge of effectively discriminating between hard negative pairs within graph data. For example, CuCo (Chu et al., 2021) applies a scoring function to rank the negative samples from easy to hard. ProGCL (Xia, Wu et al., 2022) constitutes a novel measure for the hardness of negative samples by a probability estimator. HSAN (Liu, Yang, Zhou, Liu, Wang, Liang, Tu, Li, Duan et al., 2023) proposes a weight modulating function to recognize both the hard negative and positive samples, and dynamically adjust the weights of hard and easy pairs.

However, most of these methods primarily focus on the contrast between node-node level, which neglects the exploration of high-order semantics and results in an insufficient utilization of topological information (Ju et al., 2023; Wang et al., 2022c). In this paper, we propose a multi-scale contrastive scheme to leverage the high-order structural contexts of nodes and their regional neighbors for the graph clustering tasks.

3. Problem definition and preliminaries

In this section, we first give the notations and problem definition of our paper. Then, we introduce some important concepts for the task.

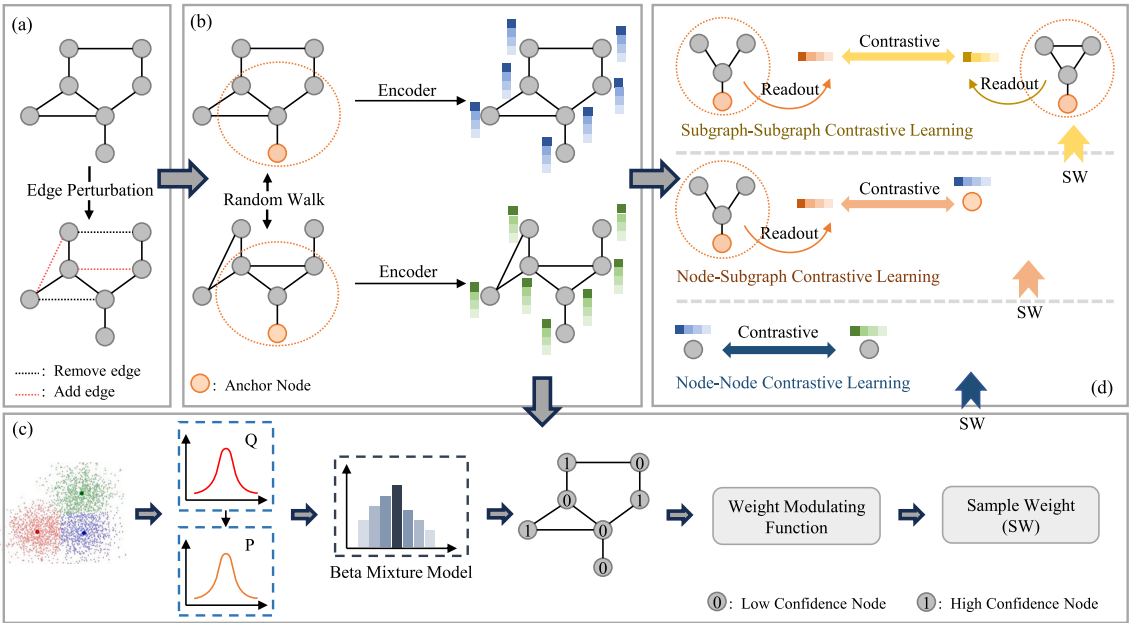

Fig. 1. An overview of our MHGC. (a) The original graph and its augmented version are treated as two views of the graph and processed by Laplacian filters. (b) The multi-scale representations, encompassing the node and the corresponding subgraph feature, of each view are embedded by attribute encoders. (c) A dual graph clustering layer is proposed to derive the soft clustering labels, and high-confidence samples are selected via the Beta Mixture Model. (d) Build upon this, the hard samples are dynamically weighted in multi-scale contrastive learning.

3.1. Notations

Let an arbitrary undirected graph be represented by a tuple $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} = \{v_1, \ldots, v_N\}$ denotes the node set with N nodes, $\mathcal{E}$ is the edge set with M edges. The node feature matrix is denoted as $X = (x_1, \ldots, x_N)^{\mathrm{T}} \in \mathbb{R}^{N \times d}$, where each row $x_i \in \mathbb{R}^d$ represents the feature vector of node i and d is the dimension of node feature. We use adjacency matrix $A = (a_{ij}) \in \mathbb{R}^{N \times N}$ to characterize the structural information of $\mathcal{G}$, which is generated according to edge connections in $\mathcal{E}$, namely, each entry $a_{ij} = 1$ if $(v_i, v_j) \in \mathcal{E}$, otherwise, $a_{ij} = 0$. The corresponding degree matrix can be formulated as $D = \mathrm{diag}(d_1, \ldots, d_N)$ with degree $d_i = \sum_{j=1}^{N} a_{ij}$. The graph Laplacian matrix is defined as $L = D - A$ and can be normalized as $\tilde{L} = I - \hat{D}^{-\frac{1}{2}} \hat{A} \hat{D}^{-\frac{1}{2}}$, where $\hat{A} = A + I$ for considering self-connections, $I \in \mathbb{R}^{N \times N}$ is the identity matrix.

3.2. Deep graph clustering

Given an unlabeled graph consisting of N nodes, the target of deep graph clustering is to partition these nodes into several disjoint groups without human annotations. Concretely, node embeddings $Z \in \mathbb{R}^{N \times d}$ are first learned in an unsupervised manner by encoding the graph structure and node attribute with a deep neural network $\mathcal{F}$, and the process can be defined as follows:

$$Z = \mathcal{F}(A, X). \tag{1}$$

Then, based on the learned embedding, a clustering algorithm (e.g. spectral clustering, clustering neural network layer, or the K-means Bo et al., 2020) is employed to divide these nodes into K groups $\{C_1, \ldots, C_K\}$.

4. Methodology

In this section, we first introduce the motivation and architecture of the proposed MHGC. We then present the details of each component and the overall optimization for deep graph clustering.

4.1. Overview

The basic idea of our proposed MHGC is to explicitly exploit high-order structure contexts with the hierarchical multi-scale contrastive scheme and focus more on hard samples to promote the learned node representation for graph clustering. As shown in Fig. 1, there are four parts in MHGC framework: Graph Augmentation, Multi-Scale Graph Encoding, Dual Self-Supervised Graph Clustering and Hard Sample Aware Contrastive Learning. In the Graph Augmentation module, we treat the original graph and its edge

modification version as two views of the graph. Then in the Multi-Scale Graph Encoding module, we sample the subgraph context of a node with random walk and encode different scales of information, encompassing the node and corresponding subgraph feature. So in this process, the high-order context information is integrated to boost the quality of the representation. After that, we perform a dual graph clustering layer on the embedded representation to derive the soft clustering labels in a self-supervised manner, and high-confidence clustering results are selected via the proposed mixture model. Finally, we place greater emphasis on hard positive and negative samples across various representation granularities and propose a dynamic weighted multi-scale contrastive scheme that leverages high-confidence clustering results and sample similarity.

4.2. Graph augmentation

Graph augmentation methods are crucial for unsupervised learning models, as they directly impact the learned node embeddings. Effective graph augmentation methods should assist the model in uncovering semantic information about nodes, enabling the discovery of deeper features directly relevant to downstream tasks like graph clustering. Here, we employ edge perturbation to augment the graph in our framework, where a subset of edges is randomly removed, and another random subset of edges is added. Then, the input graph serves as the first view, while the augmented graph forms the second view. We sample nodes and subgraphs from both views of the graph, which serve as inputs for the subsequent model training.

Edge Modification. Given the input graph $\mathcal{G}$, we employ edge perturbation as the graph augmentation method in our framework. Instead of merely dropping edges from the graph, we also add an equivalent number of edges as those dropped to the graph (Duan et al., 2023; Jin, Zheng et al., 2021). Specifically, given the adjacency matrix A with M edges, we define an edge perturbation ratio p to randomly drop $\frac{pM}{2}$ edges from A and then randomly adding $\frac{pM}{2}$ of new edges to A. Notice that both the edge dropping and adding processes follow an i.i.d. uniform distribution. In this way, the edge modification approach preserves the original properties of the graph while complicating the augmented view without compromising node semantics.

Random Walk Subgraph Sampling. Since the node feature in the graph is related to its surrounding neighbors, we explicitly exploit the high-order structure contexts of the node to promote the node representation learning for the graph clustering task. Inspired by the previous work (Qiu et al., 2020), we adopt random walk with restart (RWR) (Tong et al., 2006) to sample the ego subgraph around the target node, which enables to capture the structure and attribute information from the surrounding neighbors. Specifically, we start a random walk on the graph from the ego node v_i by iteratively traveling to its neighbors and returning back with a probability proportional to the edge weight. Then, the walked nodes are collected as a subset S_i and regarded as an ego subgraph of node v_i.

4.3. Multi-scale graph encoding

We design an attribute graph encoder to map the nodes and subgraphs into the hidden space. Following the previous work (Cui et al., 2020), we initially leverage Laplacian filter to aggregate neighbor information and eliminate the high-frequency noises from the attribute matrix, defined as follows:

$$\tilde{X} = (\prod_{i=1}^{t}(1-\tilde{L}))X = (1-\tilde{L})^t X, \tag{2}$$

where $(1-\tilde{L})$ is the graph Laplacian filter stacking up t times to achieve loss-pass filtering. $\tilde{X}$ is the corresponding filtered node feature matrix. Thus, given two input graph views, namely the original graph $\mathcal{G}$ and its edge perturbation version $\mathcal{G}'$, we apply the Laplacian filter to obtain the corresponding smoothed attribute matrix $\tilde{X}_1$ and $\tilde{X}_2$. Then, we encode $\tilde{X}_1$ and $\tilde{X}_2$ with two encoders AE_1 and AE_2 as follows:

$$\begin{aligned} Z_1 &= (z_1^{v_1}, \ldots, z_1^{v_N}) = \mathrm{AE}_1(\tilde{X}_1);\ z_1^{v_i} = \frac{z_1^{v_i}}{\|z_1^{v_i}\|_2}, i = 1, \ldots, N, \\ Z_2 &= (z_2^{v_i}, \ldots, z_2^{v_N}) = \mathrm{AE}_2(\tilde{X}_2);\ z_2^{v_i} = \frac{z_2^{v_i}}{\|z_2^{v_i}\|_2}, i = 1, \ldots, N, \end{aligned} \tag{3}$$

where Z_1 and Z_2 are node embeddings of the two views. AE_1 and AE_2 denote the attribute encoders. In practice, we simply implement via two multi-layer perceptions (MLPs) following the previous works (Liu, Yang, Zhou, Liu, Wang, Liang, Tu, Li, Duan et al., 2023; Yang et al., 2023), which has the same structure but with unshared parameters.

For two input graph view, we sample the ego subgraph S_i of node v_i and the representation of subgraph is derived from the aggregation of all node representations within the subgraph by readout function. This can be formally described as follows:

$$\begin{aligned} z_1^{S_i} &= \mathrm{READOUT}(\{z_1^v\}_{v \in S_i^1}), \\ z_2^{S_i} &= \mathrm{READOUT}(\{z_2^v\}_{v \in S_i^2}), \end{aligned} \tag{4}$$

where $z_1^{S_i}$ is the subgraph-level representation, READOUT is implemented via the averaging pooling function.

4.4. Dual self-supervised graph clustering

Since graph clustering is deemed as an unsupervised task without available feedback, we perform a dual self-supervised clustering layer on the encoded node representation of the original input graph Z_1 to derive the pseudo soft label. In particular, for each node v_i and each cluster C_k, we measure their similarities as the soft cluster assignment $Q = (q_{ik}) \in \mathbb{R}^{N\times K}$ by utilizing the student's t-distribution as a kernel, which can be defined as:

$$q_{ik} = \frac{(1 + \|z_1^{v_i} - \mu_k\|^2)^{-1}}{\sum_{k'}(1 + \|z_1^{v_i} - \mu_{k'}\|^2)^{-1}}, \tag{5}$$

where u_k indicates cluster center vector of C_k. After obtaining the initial probability, we aim to optimize the node representation for clustering by learning from the high-confidence assignments. Thus, we foster all nodes to get closer to the corresponding cluster centers and the target distribution $P = (p_{ik}) \in \mathbb{R}^{N\times K}$ can be sharpened as:

$$p_{ik} = \frac{q_{ik}^2 / \sum_i q_{ik}}{\sum_k (q_{ik}^2 / \sum_i q_{ik})}. \tag{6}$$

In the target distribution, we square and normalize each node-cluster pair in the assignment distribution so that the cluster assignments will have higher confidence. Therefore, we further minimize the KL divergence between q_{ik} and p_{ik} to align the student's distribution to the target distribution:

$$\mathcal{L}_{clu} = \mathrm{KL}(P \parallel Q) = \sum_i \sum_k p_{ik} \log \frac{p_{ik}}{q_{ik}}. \tag{7}$$

By utilizing the target distribution P as supervision, the refined distribution Q can be further used to update P in turn. This cluster assignment process helps to obtain more discriminative node representation for graph clustering. And finally we adopt the well-trained assignment matrix Q to predict the cluster label of node, i.e., node v_i can be:

$$y_i = \mathrm{argmax}_{k\in\{1,\dots,K\}} q_{ik}. \tag{8}$$

4.5. Hard sample aware contrastive learning

The paradigm of graph contrastive learning has demonstrated its effectiveness in unsupervised settings by efficiently capturing the semantic information of nodes. As effective representation benefits the graph clustering process, we propose a multi-scale contrastive learning scheme with newly added subgraph information. Meanwhile, since hard samples usually confuse the model learning, we focus on the positive and hard negative samples and propose a hard sample aware contrastive loss with dynamic weighting strategy to further guide our contrastive framework.

Cluster-guided Hard Sample Mining. In this part, we focus on identifying hard samples for graph clustering. As our hard sample is guided by the clustering result, we first use the difference in self-supervised clustering loss to select nodes with high-confidence cluster assignments. More concretely, we utilize a two-component (high-low) Beta Mixture Model (BMM) to estimate the confidence probability. And the probability density function (pdf) of beta distribution over the per-sample clustering loss ℓ_i ($\mathcal{L}_{clu} = \sum_i \ell_i$) can be defined as follows:

$$p(\ell|\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \ell^{\alpha-1}(1-\ell)^{\beta-1}, \tag{9}$$

where $\alpha, \beta > 0$ are the parameters of beta distribution and $\Gamma(\cdot)$ represents the Gamma function. The mixture pdf of C component on ℓ can be formulated as:

$$p(\ell) = \sum_{c=1}^{C} \lambda_c p(\ell|\alpha_c,\beta_c), \tag{10}$$

where λ_c is the mixture coefficient. Here we let $C = 2$ to model the distribution of nodes with high and low confidence in their clustering assignments. Expectation Maximization (EM) algorithm is then applied to fit the BMM to the observed per-sample clustering loss distribution. In E-step, we fix the parameter of BMM and compute the probability of node v_i being high or low confidence sample with Bayes rule, defined as:

$$p(c|\ell_i) = \frac{\lambda_c p(\ell_i|\alpha_c,\beta_c)}{\sum_{c'=1}^{C} \lambda_{c'} p(\ell_i|\alpha_{c'},\beta_{c'})}. \tag{11}$$

Then for M-step, we fixed $p(c|\ell_i)$ and estimate parameters α_c and β_c using the method of moments in statistics:

$$\alpha_c = \tilde{\ell}_c\left(\frac{\tilde{\ell}_c(1-\tilde{\ell}_c)}{s_c^2} - 1\right),\ \beta_c = \frac{\alpha_c(1-\tilde{\ell}_c)}{\tilde{\ell}_c}, \tag{12}$$

where $\tilde{\ell}_c$ and s_c^2 are the weighted average of the per-sample clustering loss and corresponding variance estimate:

$$\tilde{\ell}_c = \frac{\sum_{i=1}^N p(c|\ell_i)\ell_i}{\sum_{i=1}^N p(c|\ell_i)},\ s_c^2 = \frac{\sum_{i=1}^N p(c|\ell_i)(\ell_i - \tilde{\ell}_c)^2}{\sum_{i=1}^N p(c|\ell_i)}. \tag{13}$$

And the mixing coefficient λ_c can be computed as:

$$\lambda_c = \frac{1}{N}\sum_{i=1}^{N} p(c|\ell_i). \tag{14}$$

The E and M-steps are performed iteratively until convergence or reaching the maximum number of iterations (i.e., 10 in our experiments). By setting a threshold τ on the posterior probability $p(c|\ell_i)$, where high-confidence one are with lower mean, we select the nodes set $\mathcal{H}$ with high-confidence cluster assignments.

In this way, nodes within the same clusters are regarded as positive pairs while those from distinct clusters are recognized as possible negative pairs. As a result, negative pairs with high similarity and positive pairs with low similarity are identified as hard samples and the weight can be defined as:

$$\omega(v_i, v_j) = \begin{cases} 1, & \neg(i, j \in \mathcal{H}), \\ |r_{ij} - \mathrm{Norm}(s(v_i, v_j))|^{\gamma}, & \text{otherwise}, \end{cases} \tag{15}$$

where r_{ij} reveals the clustering relationship between node v_i and v_j, namely $r_{ij} = 1$ if $y_i = y_j$, otherwise $r_{ij} = 0$. $s(v_i, v_j)$ denotes the similarity between the node pair and $\mathrm{Norm}(\cdot)$ is the min–max normalization. γ is the focusing factor used to control the rate at which the sample is weighted.

Hard-aware Multi-scale Contrast Scheme. We consider a hierarchical multi-scale contrastive learning framework, which includes three types, node-subgraph, node-node and subgraph-subgraph contrast. Meanwhile, instead of treating each sample pair equally, we focus more on the selected hard nodes across different scales to uncover the intrinsic discriminative structure information for graph clustering.

Node-subgraph contrast. Node-subgraph contrast focuses on identifying inherent patterns from the node perspective within each graph view. And the learning objective is to enhance the alignment between the node embedding and its corresponding ego-subgraph, i.e., for the original input graph view:

$$\mathcal{L}_{ns}^{1} = \sum_{i} -\log\left(\frac{e^{\omega(v_i^1, S_i^1)\cdot s(v_i^1, S_i^1)}}{e^{\omega(v_i^1, S_i^1)\cdot s(v_i^1, S_i^1)} + \sum_{j\neq i} e^{\omega(v_i^1, S_j^1)\cdot s(v_i^1, S_j^1)}}\right), \tag{16}$$

where $s(v_i^1, S_i^1) = (z_1^{v_i})^{\mathrm{T}} z_1^{S_i}$ denotes the similarity between node and subgraph embedding, $\omega(v_i^1, S_i^1)$ is the corresponding weight focusing more on the hard samples. Dynamically adjusting the weights during training allows the chosen hard samples to effectively convey their neighboring structures in the latent space. Similarly, we can also obtain $\mathcal{L}_{ns}^{2}$ and the final contrast loss is:

$$\mathcal{L}_{ns} = \frac{1}{2N}\sum_{j=1}^{2} \mathcal{L}_{ns}^{j} \tag{17}$$

Node-node contrast. Node-node contrast is capable of effectively capturing information at the node level, enhancing the expressiveness of the encoded node embeddings. We contrast the node embeddings that span across two views. Thus, the same node from the different views forms a positive pair while different nodes form a negative pair. The loss function can be then defined as:

$$\mathcal{L}_{nn}^{1} = \sum_{i} -\log\left(\frac{e^{\omega(v_i^1, v_i^2)\cdot s(v_i^1, v_i^2)}}{e^{\omega(v_i^1, v_i^2)\cdot s(v_i^1, v_i^2)} + \sum_{j\neq i} e^{\omega(v_i^1, v_j^2)\cdot s(v_i^1, v_j^2)}}\right), \tag{18}$$

where $s(v_i^1, v_i^2) = (z_1^{v_i})^{\mathrm{T}} z_2^{v_i}$ and $\omega(v_i^1, v_i^2)$ denote the similarity and the up/down-weighting rate of hard/easy samples. Note that we can also calculate $\mathcal{L}_{nn}^{2}$, and the node-node contrast loss can be calculated as $\mathcal{L}_{nn}$.

Subgraph-subgraph contrast. Subgraph-subgraph contrast aims to incorporate more representative and intrinsic subgraph embedding into the graph clustering task. We treat the subgraph as forming a positive pair with its perturbed version, which is the ego-subgraph of the same target node v_i appearing in another view. To this end, the subgraph-subgraph contrast loss can be defined as:

$$\mathcal{L}_{ss}^{1} = \sum_{i} -\log\left(\frac{e^{\omega(S_i^1, S_i^2)\cdot s(S_i^1, S_i^2)}}{e^{\omega(S_i^1, S_i^2)\cdot s(S_i^1, S_i^2)} + \sum_{j\neq i} e^{\omega(S_i^1, S_j^2)\cdot s(S_i^1, S_j^2)}}\right), \tag{19}$$

where $s(S_i^1, S_i^2) = (z_1^{S_i})^{\mathrm{T}} z_2^{S_i}$ and $\omega(S_i^1, S_i^2)$ denote the similarity and the corresponding adjust weight. Similarly, we can also calculate $\mathcal{L}_{ss}^{2}$ and get $\mathcal{L}_{ss}$.

4.6. Optimization target

To integrate the advantages of the three contrasts, we jointly optimize three different levels of the contrast loss function with the self-supervised clustering loss. The total loss of the framework can be:

$$\mathcal{L} = \kappa_1 \mathcal{L}_{ns} + \kappa_2 \mathcal{L}_{nn} + \kappa_3 \mathcal{L}_{ss} + \mathcal{L}_{clu}, \tag{20}$$

where κ represents the hyper-parameter to trade off the weight of contrastive loss and self-supervised clustering loss. The detailed training process of our MHGC is illustrated in Algorithm 1.

Algorithm 1 The overall training process of MHGC

Input: Graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with adjacency matrix A and feature matrix X; cluster number K.
Output: The final clustering result Y.

while not convergence **do**
  Form two graph views from the original graph and the edge modification version. Sample subgraphs via RWR in each view; Divide $\mathcal{V}$ into batches $\mathcal{B}$ by random;
  for $v_i \in \mathcal{B}$ **do**
    Encode multi-scale graph information to get node and subgraph representation $z_1^{v_i}, z_2^{v_i}, z_1^{S_i}, z_2^{S_i}$; // Eqs. (3)-(4)
    Calculate cluster assignment distribution Q and P, and get the self-supervised clustering loss $\mathcal{L}_{clu}$; // Eqs. (5)-(7)
    Obtain high-confidence clustering results via fitting BMM and select hard sample pairs ; // Eqs. (9)-(15)
    Perform hard-aware multi-scale contrastive learning via down(up)-weighting easy(hard) samples ; // Eqs. (16)-(19)
  end for
  Obtain the clustering result Y with well-trained Q
end while

Table 1
The statistics of datasets.

Datasets	Nodes	Edges	Attributes	Classes
Cora	2708	5429	1433	7
Citeseer	3327	4732	3703	6
AMAP	7650	119 081	745	8
BAT	131	1038	81	4
EAT	399	5994	203	4
UAT	1190	13 599	239	4

5. Experiment

In this section, we carry out comprehensive experiments on six commonly used benchmark datasets to assess the effectiveness of our MHGC and address the following research problems:

- **RQ1:** How does our proposed MHGC performs compared with baselines for graph clustering?
- **RQ2:** How do the multi-scale contrastive scheme and BMM-based hard negative samples mining strategy influence the model performance?
- **RQ3:** How do different hyper-parameters in MHGC influence the model performance?
- **RQ4:** Can we intuitively visualize the effectiveness of MHGC?

5.1. Experimental setup

Datasets. Following the setting of previous works (Liu, Yang, Zhou, Liu, Wang, Liang, Tu, Li, Duan et al., 2023; Yang et al., 2023), we evaluate the performance of MHGC on six benchmark graph clustering datasets, including CORA (McCallum et al., 2000), CITE (Giles et al., 1998), AMAP (McAuley et al., 2015), BAT, EAT, and UAT (Mrabah et al., 2022). These datasets span various types of networks, including citation networks, social networks, and traffic networks, exhibiting significant differences in scale. They serve as generalized benchmarks widely used in prior graph clustering research. We briefly introduce the mentioned datasets below and the statistics of these datasets are listed in Table 1.

- **CORA** contains 2708 papers across seven topics: algorithm, theory, neural network, case-based, probabilistic method, genetic, and reinforcement learning, where nodes denote published papers and edges denote citation relationships.
- **CITE** contains 3327 scientific papers on six topics: machine learning, database, agent, information retrieval, human-computer interaction, and artificial intelligence. Similarly to CORA, nodes denote papers and edges represent citation relationships.
- **AMAP** comprises 7,650 products categorized into eight distinct groups, based on the Amazon co-purchase graph, where nodes symbolize products and edges indicate the frequent purchasing relationships between pairs of items.
- **BAT, EAT and UAT** are air-traffic datasets sourced from the National Civil Aviation Agency of Brazil, Europe, and the Bureau of Transportation Statistics of the U.S.. Each node in these databases represents an airport, representing airport networks. The four groups into which the airports are divided reflect the importance or activity level of each airport. Furthermore, the network's edges show that there are commercial flights operating between the airports.

Baselines. We compare MHGC against eleven state-of-the-art deep graph clustering methods with their original settings. These methods can be grouped into three categories, including classical deep graph clustering methods (*i.e.,* DAEGC Wang et al., 2019, SDCN Bo et al., 2020, DFCN Tu et al., 2021), contrastive deep graph clustering methods (*i.e.,* AutoSSL Jin, Liu et al., 2021, AFGRL

Table 2
The mean performance across six benchmark datasets during ten runs. Four measures are used to evaluate the performance, and the mean value and standard deviation are presented. The top results are indicated by red values, while the runner-up results are indicated by blue values.

Dataset	Metric	Classical deep graph clustering			Contrastive deep graph clustering				Hard sample mining			
		DAEGC IJCAI 19	SDCN WWW 20	DFCN AAAI 21	AutoSSL ICLR 22	AFGRL AAAI 22	CCGC AAAI 23	MAGI KDD 24	GDCL IJCAI 21	ProGCL ICML 22	HSAN AAAI 23	MHGC Ours
CORA	ACC	70.43 ± 0.4	35.60 ± 2.8	36.33 ± 0.5	63.81 ± 0.6	26.25 ± 1.2	73.88 ± 1.2	75.72 ± 2.1	70.83 ± 0.5	57.13 ± 1.2	77.07 ± 1.6	77.72 ± 0.4
	NMI	52.89 ± 0.7	14.28 ± 1.9	19.36 ± 0.9	47.62 ± 0.5	12.36 ± 1.5	56.45 ± 1.0	55.07 ± 0.7	56.60 ± 0.4	41.02 ± 1.3	59.21 ± 1.0	57.50 ± 1.0
	ARI	49.63 ± 0.4	07.78 ± 3.2	04.67 ± 2.1	38.92 ± 0.8	14.32 ± 1.9	52.51 ± 1.9	57.34 ± 2.4	48.05 ± 0.7	30.71 ± 2.7	57.52 ± 2.7	57.96 ± 1.5
	F1	68.27 ± 0.6	24.37 ± 1.0	26.16 ± 0.5	56.42 ± 0.2	30.20 ± 1.2	70.98 ± 2.8	73.91 ± 1.4	52.88 ± 1.0	45.68 ± 1.3	75.11 ± 1.4	75.50 ± 1.1
CITE	ACC	64.54 ± 1.4	65.96 ± 0.3	69.50 ± 0.2	66.76 ± 0.7	31.45 ± 0.5	69.84 ± 0.9	70.60 ± 0.6	66.39 ± 0.7	65.92 ± 0.8	71.15 ± 0.8	72.21 ± 0.7
	NMI	36.41 ± 0.9	38.71 ± 0.3	43.90 ± 0.2	40.67 ± 0.8	15.17 ± 0.5	44.33 ± 0.8	43.14 ± 0.5	39.52 ± 0.4	39.59 ± 0.4	45.06 ± 0.7	45.88 ± 0.8
	ARI	37.78 ± 1.2	40.17 ± 0.4	45.50 ± 0.3	38.73 ± 0.6	14.32 ± 0.8	45.68 ± 1.8	46.87 ± 0.7	41.07 ± 1.0	36.16 ± 1.1	47.05 ± 1.1	48.34 ± 0.8
	F1	62.20 ± 1.3	63.62 ± 0.2	64.30 ± 0.2	58.22 ± 0.7	30.20 ± 0.7	62.71 ± 2.1	62.81 ± 0.4	61.12 ± 0.7	57.89 ± 2.0	63.01 ± 1.8	63.48 ± 1.1
AMAP	ACC	75.96 ± 0.2	53.44 ± 0.8	76.82 ± 0.2	54.55 ± 1.0	75.51 ± 0.8	77.25 ± 0.4	77.40 ± 0.3	43.75 ± 0.8	51.53 ± 0.4	77.02 ± 0.3	79.24 ± 0.5
	NMI	65.25 ± 0.5	44.85 ± 0.8	66.23 ± 1.2	48.56 ± 0.7	64.05 ± 0.2	67.44 ± 0.5	67.85 ± 0.3	37.32 ± 0.3	39.56 ± 0.4	67.21 ± 0.3	68.79 ± 0.3
	ARI	58.12 ± 0.2	31.21 ± 1.2	58.28 ± 0.7	26.87 ± 0.3	54.45 ± 0.5	57.99 ± 0.7	57.63 ± 0.6	21.57 ± 0.5	34.18 ± 0.9	58.01 ± 0.5	62.67 ± 0.7
	F1	69.87 ± 0.5	50.66 ± 1.5	71.25 ± 0.3	54.47 ± 0.8	69.99 ± 0.3	72.18 ± 0.6	71.79 ± 0.3	38.37 ± 0.3	31.97 ± 0.4	72.03 ± 0.5	77.12 ± 0.4
BAT	ACC	52.67 ± 0.0	53.05 ± 4.6	55.73 ± 0.1	42.43 ± 0.5	50.92 ± 0.4	75.04 ± 1.8	40.61 ± 0.9	45.42 ± 0.5	55.73 ± 0.8	77.15 ± 0.7	78.41 ± 0.6
	NMI	21.43 ± 0.4	25.74 ± 5.7	48.77 ± 0.5	17.84 ± 1.0	27.55 ± 0.6	50.23 ± 0.4	13.56 ± 1.6	31.70 ± 0.4	28.69 ± 0.9	53.21 ± 0.9	54.27 ± 0.7
	ARI	18.18 ± 0.3	21.04 ± 5.0	37.76 ± 0.2	13.11 ± 0.8	21.89 ± 0.7	46.95 ± 3.1	06.71 ± 0.8	19.33 ± 0.6	21.84 ± 1.3	52.20 ± 1.1	52.38 ± 0.7
	F1	52.23 ± 0.0	46.45 ± 5.9	50.90 ± 0.1	34.84 ± 0.2	46.53 ± 0.6	74.90 ± 1.8	40.93 ± 0.9	39.94 ± 0.6	56.08 ± 0.9	77.13 ± 0.8	78.38 ± 0.6
EAT	ACC	36.89 ± 0.2	39.07 ± 1.5	49.37 ± 0.2	31.33 ± 0.5	37.42 ± 1.2	57.19 ± 0.7	35.04 ± 1.1	33.46 ± 0.2	43.36 ± 0.9	56.69 ± 0.3	58.48 ± 0.7
	NMI	05.57 ± 0.1	08.83 ± 2.5	32.90 ± 0.4	07.63 ± 0.9	11.44 ± 1.4	33.85 ± 0.9	06.07 ± 0.5	13.22 ± 0.3	23.93 ± 0.5	33.25 ± 0.4	34.47 ± 0.6
	ARI	05.03 ± 0.1	06.31 ± 2.0	23.25 ± 0.2	02.13 ± 0.7	06.57 ± 1.7	27.71 ± 0.4	04.11 ± 0.5	04.31 ± 0.3	15.03 ± 1.0	26.85 ± 0.6	28.18 ± 0.8
	F1	34.72 ± 0.2	33.42 ± 3.1	42.95 ± 0.0	21.82 ± 1.0	30.53 ± 1.5	57.09 ± 0.9	35.04 ± 1.1	25.02 ± 0.2	42.54 ± 0.5	57.26 ± 0.3	58.54 ± 0.6
UAT	ACC	52.29 ± 0.5	52.25 ± 1.9	33.61 ± 0.1	42.52 ± 0.6	41.50 ± 0.3	56.34 ± 1.1	45.95 ± 1.5	48.70 ± 0.1	45.38 ± 0.6	56.04 ± 0.7	56.72 ± 0.4
	NMI	21.33 ± 0.4	21.61 ± 1.3	26.49 ± 0.4	17.86 ± 0.2	17.33 ± 0.5	28.15 ± 1.9	12.91 ± 0.6	25.10 ± 0.0	22.04 ± 2.2	26.99 ± 2.1	27.39 ± 0.6
	ARI	20.50 ± 0.5	21.63 ± 1.5	11.87 ± 0.2	13.13 ± 0.7	13.62 ± 0.6	25.52 ± 2.1	12.54 ± 0.6	21.76 ± 0.0	14.74 ± 2.0	25.22 ± 2.0	25.72 ± 0.8
	F1	50.33 ± 0.6	45.59 ± 3.5	25.79 ± 0.3	34.94 ± 0.9	36.52 ± 0.9	55.24 ± 1.7	45.95 ± 1.5	45.69 ± 0.1	39.30 ± 1.8	54.20 ± 1.8	55.61 ± 1.1
	Avg. rank	6.92	7.71	5.92	8.71	8.54	3.21	6.29	7.54	7.33	2.63	1.17

Lee et al., 2022, CCGC Yang et al., 2023, MAGI Liu et al., 2024), and hard sample mining methods (*i.e.,* GDCL Zhao et al., 2021, ProGCL Xia, Wu et al., 2022, HSAN Liu, Yang, Zhou, Liu, Wang, Liang, Tu, Li, Duan et al., 2023).

Evaluation Metrics. Following the previous research (Liu, Yang, Zhou, Liu, Wang, Liang, Tu, Li, Duan et al., 2023; Yang et al., 2023), we adopt ACC, NMI, ARI, and F1 to measure the graph clustering performance of our model. We conduct experiments 10 times with random seeds for each dataset and record the mean and the standard deviation of these for metrics.

Implementation. In practice, we set the maximum training epoch to 400 and adopt Adam optimizer (Kingma & Ba, 2015) to minimize the total loss. We set the perturbation rate p to a fixed value 0.2. We fix the confidence threshold τ to 0.2, the focusing factor γ to 2, and the influence factors of node-node contrastive loss, node-subgraph contrastive loss, and subgraph-subgraph contrastive loss to 0.8, 0.2, and 0.1, respectively. Due to the variation in the size of datasets and unknown inherent structures, we search the optimal values of Laplacian filtering t and subgraph sizes s from the range $[2, 3, 4, 5, 6, 7, 8]$ and $[2, 4, 6, 8, 10]$, respectively.

5.2. Performance comparison (RQ1)

We evaluate the performance of our MHGC compared with all baselines and summarize the results in Table 2. From the experimental results, we draw the following conclusions:

- Compared to classical deep graph clustering methods, most contrastive learning approaches show improved performance, which implicitly incorporate supervision information from different graph views, effectively mitigating the impact of noise or low-quality input data to produce high-quality node representations.
- Most hard-aware contrastive learning methods, such as HSAN and our proposed MHGC, consistently outperform traditional contrastive deep graph clustering methods. This result highlights the strategy is essential for identifying hard negative pairs, especially those challenging yet important boundary samples, to enhance the model's discrimination capability.
- Overall, our MHGC achieves significantly improved performance compared to existing approaches. Since existing hard sample mining methods often overlook high-order structural information, we conduct hard boundary sample mining in a hierarchically contrastive manner, leading to the outstanding discriminative capability for graph clustering task.

5.3. Ablation study (RQ2)

Our proposed MHGC incorporates multi-scale graph contrastive scheme at three levels (*i.e.,* node-node level, node-subgraph level, and subgraph-subgraph level). To demonstrate the effectiveness of each component, we perform experiments on all possible combinations of these levels. Additionally, we conduct ablation studies on the vanilla version of MHGC (without the hard negative sample mining module denoted as "H") to further verify the influence of the BMM-based hard sample mining strategy. The experimental results on benchmarks are summarized in Table 3, yielding the following conclusions:

- The performance varies across the three levels. Generally, the inclusion of the node-node level or node-subgraph level tends to outperform the subgraph-subgraph level. This is mainly attributed to the fact that the contrastive learning in node-node level can capture pairwise relationships between nodes, which facilitates the learning of effective discriminative patterns for node clustering. Meanwhile, the node-subgraph level contrastive learning draws structural information from regional neighbors and provides high-order semantics for the clustering task. However, subgraph-subgraph level focuses on entire subgraphs and may ignore the fine-grained node features, leading to deficient representations for clustering.
- Nevertheless, jointly executing contrastive learning from multi-scale graph levels leads to improved performance. This highlights the significance of multi-scale contrastive learning in enriching high-order structural contexts which contributes to overall performance enhancement. These findings also suggest that properly up-weighting $\mathcal{L}_{NN}$ and $\mathcal{L}_{NS}$ is more conducive to achieving better performance.
- Additionally, it is evident that the hard sample mining technique based on BMM enhances the model's performance. This suggests that by dynamically up-weighting the learning method, the strategy directs our model to concentrate on hard sample pairs, improving the samples' discriminative power.

5.4. Parameter sensitivity (RQ3)

We then examine the sensitivity of the proposed MHGC to various hyper-parameter. Concretely, we study the effect of perturbation rate p, subgraph size $|\mathcal{S}|$, focusing factor γ for modulating the weights of hard and easy sample pairs, and confidence threshold τ for the identification of high-confidence samples.

Efficient of the Perturbation Rate p and Subgraph Size $|\mathcal{S}|$. To investigate the optimal scale of subgraph and graph augmentation, we study the performance of model with varying values of p and $|\mathcal{S}|$. Specifically, we search the values of p and $|\mathcal{S}|$ in the range of $\{0.1, 0.2, 0.4, 0.6, 0.8\}$ and $\{2, 4, 6, 8, 10\}$, respectively. Fig. 2 shows the performance w.r.t. different values of p and s on the BAT, EAT, and UAT datasets. We find that:

Table 3
Ablation study on multi-scale contrastive learning and hard sample mining strategy. The red values represent the best results, while the blue values indicate the runner-up outcomes.

Dataset	$\mathcal{L}_{NN}$	$\mathcal{L}_{NS}$	$\mathcal{L}_{SS}$	H	ACC	NMI	ARI	F1
CORA	✓	–	–	–	72.63 ± 1.6	55.14 ± 1.2	51.96 ± 2.1	68.27 ± 3.1
	–	✓	–	–	72.73 ± 1.3	55.06 ± 0.8	51.21 ± 1.8	68.69 ± 2.7
	–	–	✓	–	72.01 ± 1.5	55.12 ± 1.0	51.44 ± 2.4	67.67 ± 2.7
	✓	✓	–	–	74.38 ± 1.2	55.53 ± 1.2	53.06 ± 1.7	68.42 ± 2.6
	✓	–	✓	–	73.56 ± 1.6	54.90 ± 1.2	51.01 ± 1.4	67.85 ± 1.6
	–	✓	✓	–	73.87 ± 1.8	55.51 ± 1.1	52.51 ± 2.2	69.50 ± 2.6
	✓	✓	✓	–	75.44 ± 1.3	56.69 ± 1.3	54.45 ± 1.7	72.68 ± 1.1
	✓	✓	✓	✓	77.72 ± 0.4	57.50 ± 1.0	57.96 ± 1.5	75.50 ± 1.1
CITE	✓	–	–	–	70.61 ± 0.7	44.44 ± 0.3	46.93 ± 1.3	62.35 ± 0.4
	–	✓	–	–	70.57 ± 0.7	44.29 ± 0.4	46.84 ± 1.4	62.25 ± 0.5
	–	–	✓	–	70.52 ± 0.7	44.25 ± 0.4	46.74 ± 1.3	62.22 ± 0.5
	✓	✓	–	–	71.27 ± 0.9	45.61 ± 0.7	47.04 ± 1.4	62.53 ± 0.5
	✓	–	✓	–	71.08 ± 0.7	45.54 ± 0.4	47.04 ± 1.3	62.35 ± 0.4
	–	✓	✓	–	71.16 ± 0.7	45.38 ± 0.4	46.88 ± 1.5	62.31 ± 0.5
	✓	✓	✓	–	71.68 ± 0.7	45.61 ± 0.4	47.00 ± 1.4	63.33 ± 0.5
	✓	✓	✓	✓	72.21 ± 0.7	45.88 ± 0.8	48.34 ± 0.8	63.48 ± 1.1
AMAP	✓	–	–	–	78.14 ± 1.1	66.54 ± 0.7	60.15 ± 2.6	75.12 ± 2.3
	–	✓	–	–	78.36 ± 1.3	67.30 ± 0.6	60.66 ± 1.2	72.33 ± 1.6
	–	–	✓	–	75.68 ± 1.3	67.78 ± 1.4	55.85 ± 0.9	71.22 ± 2.1
	✓	✓	–	–	78.83 ± 0.9	68.14 ± 0.5	61.78 ± 1.8	75.78 ± 0.4
	✓	–	✓	–	77.40 ± 0.4	67.08 ± 0.4	60.38 ± 1.2	73.08 ± 0.5
	–	✓	✓	–	78.17 ± 0.2	67.79 ± 0.4	60.83 ± 0.3	72.03 ± 0.3
	✓	✓	✓	–	78.69 ± 0.9	68.69 ± 0.5	61.54 ± 1.9	75.27 ± 0.6
	✓	✓	✓	✓	79.24 ± 0.5	68.79 ± 0.3	62.67 ± 0.7	77.12 ± 0.4
BAT	✓	–	–	–	75.83 ± 0.8	51.34 ± 0.5	47.16 ± 0.8	76.05 ± 0.8
	–	✓	–	–	75.96 ± 0.6	51.31 ± 0.6	47.59 ± 1.0	76.11 ± 0.7
	–	–	✓	–	74.62 ± 1.3	49.79 ± 1.8	45.69 ± 2.2	74.76 ± 1.3
	✓	✓	–	–	76.08 ± 1.1	50.98 ± 0.5	47.26 ± 1.3	76.31 ± 1.1
	✓	–	✓	–	76.03 ± 0.8	51.16 ± 0.5	47.37 ± 0.8	76.24 ± 0.8
	–	✓	✓	–	76.34 ± 1.1	51.39 ± 0.5	47.70 ± 1.3	76.56 ± 1.1
	✓	✓	✓	–	77.10 ± 0.7	52.88 ± 0.8	49.22 ± 1.1	77.31 ± 0.7
	✓	✓	✓	✓	78.41 ± 0.6	54.27 ± 0.7	52.38 ± 0.7	78.38 ± 0.6
EAT	✓	–	–	–	56.89 ± 1.1	33.14 ± 0.8	26.28 ± 0.9	56.30 ± 1.4
	–	✓	–	–	56.47 ± 0.9	32.68 ± 0.6	25.45 ± 1.0	56.76 ± 0.8
	–	–	✓	–	56.39 ± 0.7	33.18 ± 0.9	25.48 ± 0.7	56.44 ± 1.1
	✓	✓	–	–	57.14 ± 1.1	33.24 ± 0.9	26.12 ± 0.9	56.35 ± 1.5
	✓	–	✓	–	57.01 ± 1.1	33.12 ± 0.8	26.24 ± 0.8	56.57 ± 1.4
	–	✓	✓	–	57.39 ± 0.8	33.36 ± 0.6	26.04 ± 0.9	56.92 ± 0.7
	✓	✓	✓	–	57.89 ± 1.1	33.64 ± 0.9	27.12 ± 0.9	57.35 ± 1.5
	✓	✓	✓	✓	58.48 ± 0.7	34.47 ± 0.6	28.18 ± 0.8	58.54 ± 0.6
UAT	✓	–	–	–	55.97 ± 0.4	26.80 ± 1.1	23.68 ± 1.9	54.46 ± 1.2
	–	✓	–	–	55.38 ± 0.8	25.85 ± 1.3	24.00 ± 1.1	53.70 ± 1.1
	–	–	✓	–	55.24 ± 0.4	24.69 ± 0.5	21.66 ± 0.5	54.88 ± 0.3
	✓	✓	–	–	56.05 ± 0.4	25.94 ± 0.7	23.90 ± 1.1	55.11 ± 1.0
	✓	–	✓	–	56.22 ± 0.4	27.18 ± 0.6	23.73 ± 0.6	55.34 ± 0.5
	–	✓	✓	–	56.08 ± 0.7	26.98 ± 1.1	24.42 ± 1.2	54.10 ± 1.0
	✓	✓	✓	–	56.62 ± 0.4	26.94 ± 0.5	25.22 ± 1.4	55.65 ± 1.0
	✓	✓	✓	✓	56.72 ± 0.4	27.39 ± 0.6	25.72 ± 0.8	55.61 ± 1.1

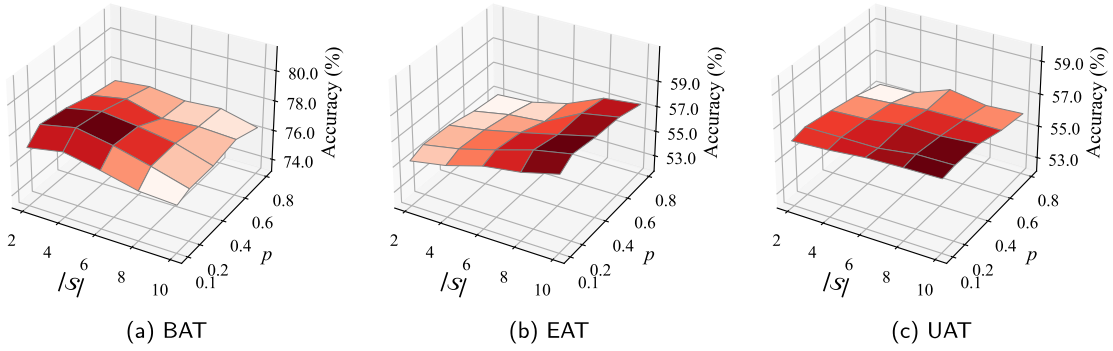

Fig. 2. Performance comparison w.r.t. different values of the hyper-parameter p and $|S|$ on the BAT, EAT, and UAT datasets.

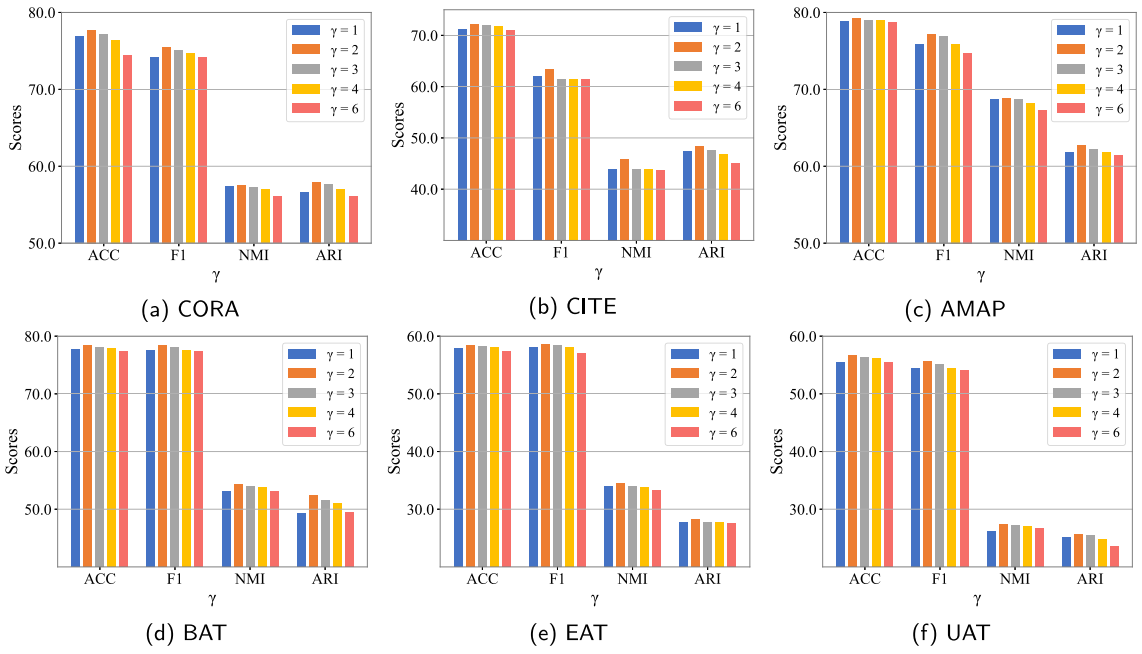

Fig. 3. Performance comparison w.r.t. different values of the hyper-parameter γ on six benchmarks.

- The optimal value of subgraph size varies across datasets. For instance, the best performance is achieved on BAT when $|S| = 4$, whereas on EAT and UAT, the best performance is achieved when $|S| = 10$. This discrepancy is due to the relatively smaller scale of BAT compared to the others. A larger value of subgraph size is more likely to introduce noise for the anchor nodes, results in a decline in performance.
- As the value of $|S|$ increases, the rising trend becomes less pronounced for both EAT and UAT. This suggests a feasible way to strike a balance between performance and computational cost by properly reducing the subgraph size, particularly on large datasets like AMAP.
- When $p = 0.1$, the augmented graph closely resembles the original graph with only slight perturbations, which leads to a noticeable decline in performance. By setting $p = 0.2$, our model reaches the optimal performance. However, as the value of p further increases, a marginal decline in performance is observed, indicating that over-perturbation may potentially disrupt the regional contexts, albeit in a minor manner.

Efficient of the Focusing Factor γ. By varying the values of focusing factor, we explore the optimal weighting rate of high-confidence samples. Specifically, we search the value of focusing factor γ in the range of $\{1, 2, 3, 4, 6\}$. Fig. 3 shows the performance w.r.t. different values of γ for modulating the weights of hard and easy sample pairs in the high-confidence set. We have the following conclusions:

- When $\gamma = 1$, the disparity between the down-weighting rate of easy sample pairs and up-weighting rate of hard sample pairs is trivial to identify tangible discriminative patterns. By setting $\gamma = 2$, MHGC achieves the best performance, illustrating the

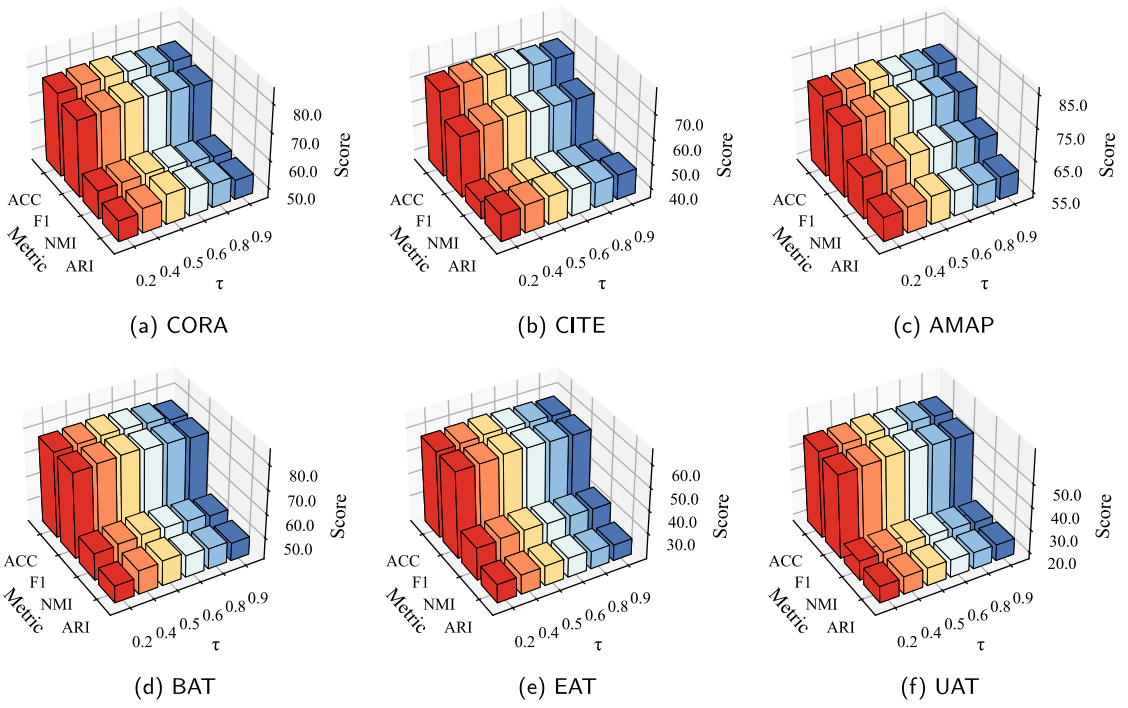

Fig. 4. Performance comparison w.r.t. different values of the hyper-parameter τ on six benchmarks.

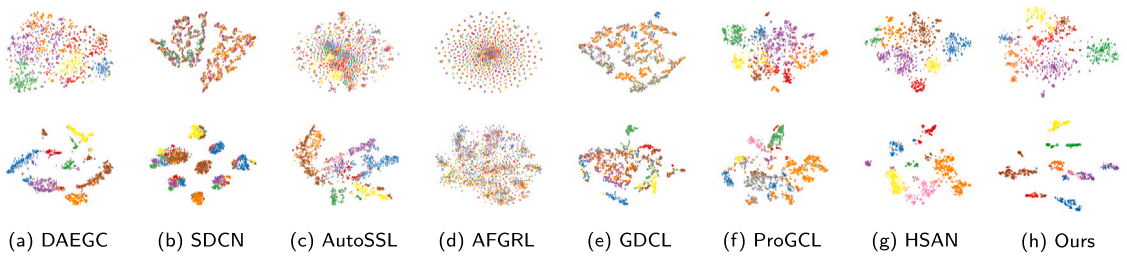

Fig. 5. 2D *t*-SNE visualization of eight methods across two benchmark datasets. The first row corresponds to the CORA dataset, while the second row pertains to the AMAP dataset.

importance of modulating the weight of hard and easy sample pairs during the training process, which significantly improves the discriminative capability.

- However, excessively increasing of γ (*i.e.*, $\gamma > 2$) results in a descending trend of model performance. This can be attributed to an excessive focusing factor, which results in an exponential decay in the down-weighting rate of easy sample pairs, negatively impacting the clustering tasks.

Efficient of the Confidence Threshold τ. We further analyze the influence of high-confidence sample set. In particular, we search the value of τ in the range of $\{0.2, 0.4, 0.5, 0.6, 0.8, 0.9\}$. Fig. 4 shows the performance w.r.t. different values of confidence threshold τ for controlling the selection of high-confidence samples. We observe that:

- When $\tau = 0.2$, the broad criteria leads to an inadequate selection of high-confidence set, causing relatively poor performance. Increasing the value of τ substantially enhances the model performance, with the best performance achieved when $\tau = 0.5$. This finding highlights the emphasis on high-confidence samples contributes to improving discriminative capability of hard pairs, thereby attaining better semantics for clustering.

- Setting the confidence threshold too high (*i.e.,* $\tau > 0.5$) results in a degradation of model performance. This is likely because the stringent criteria mistakenly identifies the high-confidence samples as low-confidence samples, which negatively impacts the learning of discriminative patterns and causes sub-optimal performance.

5.5. Case study and visualization (RQ4)

To provide an intuitive illustration of the superior performance of MHGC, we select CORA and AMAP datasets to visualize the learned node embeddings using 2D t-distributed stochastic neighbor embedding (t-SNE) (Van der Maaten & Hinton, 2008). As shown in Fig. 5, we highlight the impressive ability of our MHGC to identify discriminative patterns and reveal the underlying cluster structure. Our MHGC exhibits better separability among different clusters, characterized by improved aggregation within the same cluster and a more significant gap between different clusters, and surpass the performance of other baseline methods.

6. Conclusion

We introduce a novel contrastive-based model, referred to as MHGC, designed to identify challenging samples for the graph clustering task from a multi-scale perspective. We employ edge perturbation to generate an augmented view of the original graph and utilize random walk with restart to create subgraphs centered around each node in both views. Subsequently, an attribute encoder is utilized to embed node representations within latent space. We subsequently incorporate a clustering layer to align the learned representation with the clustering outcomes. Fitting a beta mixture model to the clustering loss enables the identification of high-confidence nodes. Utilizing high-confidence clustering information, we propose a weight regulator that dynamically increases the weights of difficult sample pairs while decreasing those of easy sample pairs in our hierarchical contrastive scheme, facilitating effective learning of discriminative patterns for the graph clustering task. Experimental results across six benchmark datasets validate the effectiveness of our MHGC. In the future, we will expand our research to include additional fields, such as node classification.

CRediT authorship contribution statement

Tao Ren: Conceptualization, Methodology, Writing – original draft, Writing – review & editing, Funding acquisition. **Haodong Zhang:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology. **Yifan Wang:** Conceptualization, Methodology, Writing – original draft, Writing – review & editing, Funding acquisition. **Wei Ju:** Writing – review & editing. **Chengwu Liu:** Methodology, Conceptualization. **Fanchun Meng:** Visualization. **Siyu Yi:** Writing – review & editing. **Xiao Luo:** Methodology, Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

The authors are grateful to the anonymous reviewers for critically reading this article and for giving important suggestions to improve this article.

Data availability

Data will be made available on request.

References

Bo, D., Wang, X., Shi, C., Zhu, M., Lu, E., & Cui, P. (2020). Structural deep clustering network. In *Proceedings of the web conference* (pp. 1400–1410).

Chen, M., Huang, C., Xia, L., Wei, W., Xu, Y., & Luo, R. (2023). Heterogeneous graph contrastive learning for recommendation. In *Proceedings of the international ACM conference on web search & data mining* (pp. 544–552).

Chu, G., Wang, X., Shi, C., & Jiang, X. (2021). CuCo: Graph representation with curriculum contrastive learning.. In *Proceedings of the international joint conference on artificial intelligence* (pp. 2300–2306).

Chuang, C.-Y., Robinson, J., Yen-Chen, L., Torralba, A., & Jegelka, S. (2020). Debiased contrastive learning. In *Proceedings of the conference on neural information processing systems* (pp. 8765–8775).

Cui, G., Zhou, J., Yang, C., & Liu, Z. (2020). Adaptive graph encoder for attributed graph embedding. In *Proceedings of the international ACM SIGKDD conference on knowledge discovery & data mining* (pp. 976–985).

Duan, Y., Liu, J., Chen, S., Chen, L., & Wu, J. (2024). G-prompt: Graphon-based prompt tuning for graph classification. *Information Processing & Management, 61*(3), Article 103639.

Duan, J., Wang, S., Zhang, P., Zhu, E., Hu, J., Jin, H., Liu, Y., & Dong, Z. (2023). Graph anomaly detection via multi-scale contrastive learning networks with augmented view. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 7459–7467).

Giles, C. L., Bollacker, K. D., & Lawrence, S. (1998). CiteSeer: An automatic citation indexing system. In *Proceedings of the third ACM conference on digital libraries* (pp. 89–98).

Gong, L., Zhou, S., Tu, W., & Liu, X. (2022). Attributed graph clustering with dual redundancy reduction. In *Proceedings of the international joint conference on artificial intelligence* (pp. 3015–3021).
Gu, Z., Luo, X., Chen, J., Deng, M., & Lai, L. (2023). Hierarchical graph transformer with contrastive learning for protein function prediction. *Bioinformatics*, *39*(7), btad410.
Halliwell, N. (2022). Evaluating explanations of relational graph convolutional network link predictions on knowledge graphs. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 12880–12881).
Hassani, K., & Khasahmadi, A. H. (2020). Contrastive multi-view representation learning on graphs. In *Proceedings of the international conference on machine learning* (pp. 4116–4126).
Hu, F., Zhu, Y., Wu, S., Wang, L., & Tan, T. (2019). Hierarchical graph convolutional networks for semi-supervised node classification. In *Proceedings of the international joint conference on artificial intelligence* (pp. 4532–4539).
Jiao, Y., Xiong, Y., Zhang, J., Zhang, Y., Zhang, T., & Zhu, Y. (2020). Sub-graph contrast for scalable self-supervised graph representation learning. In *Proceedings of the international conference on data mining* (pp. 222–231).
Jin, W., Liu, X., Zhao, X., Ma, Y., Shah, N., & Tang, J. (2021). Automated self-supervised learning for graphs. In *Proceedings of the international conference on learning representations*.
Jin, M., Zheng, Y., Li, Y.-F., Gong, C., Zhou, C., & Pan, S. (2021). Multi-scale contrastive siamese networks for self-supervised graph representation learning. In *Proceedings of the international joint conference on artificial intelligence* (pp. 1477–1483).
Ju, W., Gu, Y., Chen, B., Sun, G., Qin, Y., Liu, X., Luo, X., & Zhang, M. (2023). Glcc: a general framework for graph-level clustering. *Proceedings of the AAAI conference on artificial intelligence*, *37*(4), 4391–4399.
Ju, W., Liu, Z., Qin, Y., Feng, B., Wang, C., Guo, Z., Luo, X., & Zhang, M. (2023). Few-shot molecular property prediction via hierarchically structured learning on relation graphs. *Neural Networks*, *163*, 122–131.
Ju, W., Mao, Z., Yi, S., Qin, Y., Gu, Y., Xiao, Z., Wang, Y., Luo, X., & Zhang, M. (2024). Hypergraph-enhanced Dual Semi-supervised Graph Classification. In *Proceedings of the international conference on machine learning* (pp. 22594–22604).
Ju, W., Wang, Y., Qin, Y., Mao, Z., Xiao, Z., Luo, J., Yang, J., Gu, Y., Wang, D., Long, Q., Yi, S., Luo, X., & Zhang, M. (2024). Towards graph contrastive learning: a survey and beyond. *arXiv preprint arXiv:2405.11868*.
Ju, W., Yi, S., Wang, Y., Long, Q., Luo, J., Xiao, Z., & Zhang, M. (2024). A survey of data-efficient graph learning. In *International Joint Conference on Artificial Intelligence* (pp. 8104–8113).
Kalantidis, Y., Sariyildiz, M. B., Pion, N., Weinzaepfel, P., & Larlus, D. (2020). Hard negative mixing for contrastive learning. In *Proceedings of the conference on neural information processing systems* (pp. 21798–21809).
Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *Proceedings of the international conference on learning representations*.
Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *Proceedings of the international conference on learning representations*.
Lee, N., Lee, J., & Park, C. (2022). Augmentation-free self-supervised learning on graphs. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 7372–7380).
Li, H., Wang, Y., Xiao, Z., Yang, J., Zhou, C., Zhang, M., & Ju, W. (2024). DisCo: Graph-based disentangled contrastive learning for cold-start cross-domain recommendation. *arXiv preprint arXiv:2412.15005*.
Liu, Y., Ao, X., Feng, F., Ma, Y., Li, K., Chua, T.-S., & He, Q. (2023). FLOOD: A flexible invariant learning framework for out-of-distribution generalization on graphs. In *Proceedings of the international ACM SIGKDD conference on knowledge discovery & data mining* (pp. 1548–1558).
Liu, Y., Ding, K., Liu, H., & Pan, S. (2023). Good-d: On unsupervised graph out-of-distribution detection. In *Proceedings of the international ACM conference on web search & data mining* (pp. 339–347).
Liu, Y., Li, J., Chen, Y., Wu, R., Wang, B., Zhou, J., Tian, S., Shen, S., Fu, X., Meng, C., Wang, W., & Chen, L. (2024). Revisiting modularity maximization for graph clustering: A contrastive learning perspective. In *Proceedings of the international ACM SIGKDD conference on knowledge discovery & data mining*.
Liu, Y., Yang, X., Zhou, S., Liu, X., Wang, S., Liang, K., Tu, W., & Li, L. (2023). Simple contrastive graph clustering. *IEEE Transactions on Neural Networks and Learning Systems*, 1–12.
Liu, Y., Yang, X., Zhou, S., Liu, X., Wang, Z., Liang, K., Tu, W., Li, L., Duan, J., & Chen, C. (2023). Hard sample aware network for contrastive deep graph clustering. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 8914–8922).
Luo, J., Gu, Y., Luo, X., Ju, W., Xiao, Z., Zhao, Y., Yuan, J., & Zhang, M. (2024). GALA: Graph diffusion-based alignment with jigsaw for source-free domain adaptation. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, *46*(12), 9038–9051.
Luo, J., Luo, X., Chen, X., Xiao, Z., Ju, W., & Zhang, M. (2024). Semievol: semi-supervised fine-tuning for llm adaptation. *arXiv preprint arXiv:2410.14745*.
Ma, X., Zhang, T., & Xu, C. (2019). Gcan: Graph convolutional adversarial network for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 8266–8276).
Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, *9*(11).
Mao, Z., Ju, W., Qin, Y., Luo, X., & Zhang, M. (2023). Rahnet: retrieval augmented hybrid network for long-tailed graph classification. In *Proceedings of the 31st ACM international conference on multimedia* (pp. 3817–3826).
Mao, K., Zhu, J., Wang, J., Dai, Q., Dong, Z., Xiao, X., & He, X. (2021). Simplex: A simple and strong baseline for collaborative filtering. In *Proceedings of the international conference on information and knowledge management* (pp. 1243–1252).
McAuley, J., Targett, C., Shi, Q., & Van Den Hengel, A. (2015). Image-based recommendations on styles and substitutes. In *Proceedings of the international ACM SIGIR conference on research & development in information retrieval* (pp. 43–52).
McCallum, A. K., Nigam, K., Rennie, J., & Seymore, K. (2000). Automating the construction of internet portals with machine learning. *Information Retrieval*, *3*(2), 127–163.
Mrabah, N., Bouguessa, M., Touati, M. F., & Ksantini, R. (2022). Rethinking graph auto-encoder models for attributed graph clustering. *IEEE Transactions on Knowledge and Data Engineering*, *35*(9), 9037–9053.
Pan, S., Hu, R., Fung, S.-f., Long, G., Jiang, J., & Zhang, C. (2019). Learning graph embedding with adversarial training methods. *IEEE Transactions on Cybernetics*, *50*(6), 2475–2487.
Pan, E., & Kang, Z. (2021). Multi-view contrastive graph clustering. *Advances in Neural Information Processing Systems*, *34*, 2148–2159.
Qiu, J., Chen, Q., Dong, Y., Zhang, J., Yang, H., Ding, M., Wang, K., & Tang, J. (2020). Gcc: Graph contrastive coding for graph neural network pre-training. In *Proceedings of the international ACM SIGKDD conference on knowledge discovery & data mining* (pp. 1150–1160).
Robinson, J. D., Chuang, C.-Y., Sra, S., & Jegelka, S. (2020). Contrastive learning with hard negative samples. In *Proceedings of the international conference on learning representations*.
Roy, I., De, A., & Chakrabarti, S. (2021). Adversarial permutation guided node representations for link prediction. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 9445–9453).
Shang, B., Zhao, Y., & Liu, J. (2024). Knowledge graph representation learning with relation-guided aggregation and interaction. *Information Processing & Management*, *61*(4), Article 103752.
Tao, Z., Liu, H., Li, J., Wang, Z., & Fu, Y. (2019). Adversarial graph embedding for ensemble clustering. In *Proceedings of the international joint conference on artificial intelligence*.

Tong, H., Faloutsos, C., & Pan, J.-Y. (2006). Fast random walk with restart and its applications. In *Proceedings of the international conference on data mining* (pp. 613–622).

Tu, W., Zhou, S., Liu, X., Guo, X., Cai, Z., Zhu, E., & Cheng, J. (2021). Deep fusion clustering network. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 9978–9987).

Wang, Y., Li, Y., Li, S., Song, W., Fan, J., Gao, S., Ma, L., Cheng, B., Cai, X., & Wang, S. (2022). Deep graph mutual learning for cross-domain recommendation. In *International Conference on Database Systems for Advanced Applications* (pp. 298–305). Springer.

Wang, X., Li, J., Yang, L., & Mi, H. (2021). Unsupervised learning for community detection in attributed networks based on graph convolutional network. *Neurocomputing*, *456*, 147–155.

Wang, Y., Luo, X., Chen, C., Hua, X.-S., Zhang, M., & Ju, W. (2024). DisenSemi: Semi-supervised graph classification via disentangled representation learning. In *IEEE Transactions on Neural Networks and Learning Systems*. IEEE.

Wang, C., Pan, S., Hu, R., Long, G., Jiang, J., & Zhang, C. (2019). Attributed graph clustering: A deep attentional embedding approach. In *Proceedings of the international joint conference on artificial intelligence* (pp. 3670–3676).

Wang, C., Pan, S., Long, G., Zhu, X., & Jiang, J. (2017). Mgae: Marginalized graph autoencoder for graph clustering. In *Proceedings of the international conference on information and knowledge management* (pp. 889–898).

Wang, Y., Qin, Y., Han, Y., Yin, M., Zhou, J., Yang, H., & Zhang, M. (2022). Ad-aug: Adversarial data augmentation for counterfactual recommendation. In *Joint European conference on machine learning and knowledge discovery in databases* (pp. 474–490). Springer.

Wang, Y., Shen, J., Song, Y., Wang, S., & Zhang, M. (2022). HE-SNE: Heterogeneous event sequence-based streaming network embedding for dynamic behaviors. In *International joint conference on neural networks* (pp. 1–8). IEEE.

Wang, J., Wu, L., Zhao, H., & Jia, N. (2023). Multi-view enhanced zero-shot node classification. *Information Processing & Management*, *60*(6), Article 103479.

Wang, Y., Yang, Y., Li, S., Xie, Y., Xiao, Z., Zhang, M., & Ju, W. (2025). Gmr-rec: graph mutual regularization learning for multi-domain recommendation. *Information Sciences*, 121946.

Wu, Z., Zhan, M., Zhang, H., Luo, Q., & Tang, K. (2022). MTGCN: A multi-task approach for node classification and link prediction in graph data. *Information Processing & Management*, *59*(3), Article 102902.

Xia, W., Wang, Q., Gao, Q., Yang, M., & Gao, X. (2022). Self-consistent contrastive attributed graph clustering with pseudo-label prompt. *IEEE Transactions on Multimedia*.

Xia, W., Wang, Q., Gao, Q., Zhang, X., & Gao, X. (2021). Self-supervised graph convolutional network for multi-view clustering. *IEEE Transactions on Multimedia*, *24*, 3182–3192.

Xia, J., Wu, L., Wang, G., Chen, J., & Li, S. Z. (2022). ProGCL: Rethinking hard negative mining in graph contrastive learning. In *Proceedings of the international conference on machine learning* (pp. 24332–24346).

Xu, K., Hu, W., Leskovec, J., & Jegelka, S. (2019). How powerful are graph neural networks? In *Proceedings of the international conference on learning representations*.

Yang, X., Liu, Y., Zhou, S., Wang, S., Tu, W., Zheng, Q., Liu, X., Fang, L., & Zhu, E. (2023). Cluster-guided contrastive graph clustering network. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 10834–10842).

Yi, S., Ju, W., Qin, Y., Luo, X., Liu, L., Zhou, Y., & Zhang, M. (2023). Redundancy-free self-supervised relational learning for graph clustering. *IEEE Transactions on Neural Networks and Learning Systems*, *35*(12), 18313–18327.

Yuan, J., Hou, F., Du, Y., Shi, Z., Geng, X., Fan, J., & Rui, Y. (2022). Self-supervised graph neural network for multi-source domain adaptation. In *Proceedings of the ACM international conference on multimedia* (pp. 3907–3916).

Yuan, J., Luo, X., Qin, Y., Zhao, Y., Ju, W., & Zhang, M. (2023). Learning on Graphs under Label Noise. In *IEEE International Conference on Acoustics, Speech and Signal Processing*.

Zhang, Y., Cheng, Z., Liu, F., Yang, X., & Peng, Y. (2024). Decoupled domain-specific and domain-conditional representation learning for cross-domain recommendation. *Information Processing & Management*, *61*(3), Article 103689.

Zhang, P., Liu, X., Xiong, J., Zhou, S., Zhao, W., Zhu, E., & Cai, Z. (2020). Consensus one-step multi-view subspace clustering. *IEEE Transactions on Knowledge and Data Engineering*, *34*(10), 4676–4689.

Zhao, H., Yang, X., Wang, Z., Yang, E., & Deng, C. (2021). Graph debiased contrastive learning with joint representation clustering. In *Proceedings of the international joint conference on artificial intelligence* (pp. 3434–3440).