Deep Graph Clustering with Disentangled Representation Learning

Yifan Wang
University of International Business and Economics
Beijing, China
yifanwang@uibe.edu.cn

Yuntai Ding
Northeastern University
Shenyang, China
chinaytding@163.com

Yiyang Gu
Peking University
Beijing, China
yiyanggu@pku.edu.cn

Ziyue Qiao
Great Bay University
Dongguan, China
ziyuejoe@gmail.com

Chong Chen
Terminus Group
Beijing, China
chenchong.cz@gmail.com

Xian-Sheng Hua
Tongji University
Shanghai, China
huaxiansheng@gmail.com

Ming Zhang
Peking University
Beijing, China
mzhang_cs@pku.edu.cn

Wei Ju*
Sichuan University
Chengdu, China
juwei@scu.edu.cn

Abstract

Deep graph clustering, which aims to uncover the underlying structure within graphs and partition nodes into distinct groups, is a challenging research spot. However, the formation of the cluster in real-world graphs typically governed by the highly complex interaction of many underlying latent factors. Existing methods typically rely on the features and structure associated with the graph, and neglect the entanglement of these factors, resulting in sub-optimal clustering performance. In this paper, we propose a novel deep graph clustering framework named `DisenCluster`, which learns disentangled representations to simultaneously consider node separation results from diverse perspectives. Specifically, we introduce a disentangled graph encoder that iteratively identifies the latent factors of the input graph by modeling the distribution over different factors for each edge. Subsequently, we utilize a factor-wise contrastive loss to encourage clustering-friendly disentangled representations, allowing us to derive different clustering results based on the corresponding factor. These results are then structured as anchor graphs and seamlessly integrated into a unified graph. Finally, we formulate the framework as a continuous relaxation of the high-order graph cut problem and optimize the objective to obtain effective cluster assignments. Results from experiments on a variety of publicly available datasets further reveal the effectiveness and superiority of our `DisenCluster` compared with baselines.

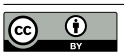
CCS Concepts

• **Computing methodologies** → **Neural networks**; **Learning latent representations**; • **Mathematics of computing** → **Graph theory**.

Keywords

Graph Neural Network; Deep Graph Clustering; Disentangled Representation Learning

*Corresponding author.



MM '25, Dublin, Ireland

ACM ISBN 979-8-4007-2035-2/2025/10
https://doi.org/10.1145/3746027.3755826

ACM Reference Format:
Yifan Wang, Yuntai Ding, Yiyang Gu, Ziyue Qiao, Chong Chen, Xian-Sheng Hua, Ming Zhang, and Wei Ju. 2025. Deep Graph Clustering with Disentangled Representation Learning. In *Proceedings of the 33rd ACM International Conference on Multimedia (MM '25), October 27–31, 2025, Dublin, Ireland.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3746027.3755826

1 Introduction

Graphs are one of the most prevalent data structures to store information about objects with complex relations and have been the subject of investigation for decades. Especially graph learning algorithms, which extract relevant features of graphs by taking advantage of machine learning algorithms, have proven enormously successful in a wide range of domains, e.g., chemical compound classification [11], social network analysis [1, 48, 52] and recommender system [21, 45, 50, 55], etc. Among these various directions in the field of graph learning, one fundamental yet challenging task, namely graph clustering, has recently garnered intensive attention [3, 24–27, 32, 39, 40, 47, 61, 64].

Built upon the superior representation power of deep learning, particularly graph neural networks (GNNs) [16, 28, 42, 57], a variety of deep graph clustering methods are successively proposed, wherein the core idea is to encode nodes through neural networks and assign them into disjoint clusters. According to the learning mechanisms, current approaches can be classified into three main categories, namely generative methods [3, 40], adversarial methods [22, 32, 39] and contrastive methods [24, 25, 61]. Generative and adversarial methods try to capture cluster-oriented node representations by recovering the graph information and distinguishing between real and synthetic samples respectively. Since most of these approaches rely on a clustering-guided loss to minimize the deviation of generated sample embeddings from pre-defined cluster centers, their performance is heavily reliant on the quality of initial cluster centers. Contrastive learning methods, on the other hand, substitute the clustering-guided loss with a more discriminative contrastive loss, offering an effective solution to mitigate the need for manual trial-and-error adjustments.

However, these existing deep graph clustering methods still remain unsatisfactory performance due to the existing limitations as follow: ❶ *Neglect feature representation entanglement.* The graph data is frequently sourced from diverse origins or acquired through

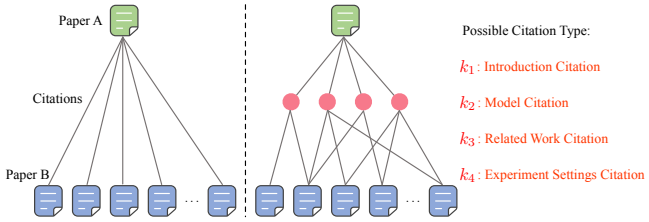


Figure 1: An illustration of diverse citation types in a citation network corresponds to different aspects of semantics between two papers, resulting in diverse clustering result structures.

various extractors, resulting in inherent variations in their feature representations. For example, the different sections within a paper in the citation network may emphasize various aspects of the paper. ❷*Unable to capture intricate semantic nuances for clustering.* The cluster formation in a real-world graph typically results from an intricate and diverse process influenced by the interplay of numerous underlying factors. We illustrate a typical example in Figure 1, where a paper in a citation network usually has edges with others for various purposes (e.g., citations of a paper scattered across different sections) [53]. Therefore, it exhibits varying node separations under each distinct aspect, concurrently, the corresponding clustering outcomes possess the potential to complement each other. Therefore, we anticipate an approach capable of investigating various underlying factors and explicitly combining node separation results from their respective diverse perspectives to enhance the performance of the clustering task.

Recently, disentangled representation learning has attracted increasing attention, which aims to acquire factorized representations that can reveal essential or explanatory aspects of the data. Additionally, disentangled representations have been shown to possess greater versatility and robustness when confronted with complex variations. In other words, each underlying factor is therefore expected to be independent and less susceptible to noise stemming from other factors [2]. In the context of graph-structured data, where numerous diverse relations are often intertwined and merged into a single graph, disentangled representation learning offers substantial advantages for downstream tasks. For graph clustering, this approach necessitates the model to generate interpretable features and disentangle the underlying relations by leveraging these features. As a result, the similarities between nodes can manifest in distinct ways under various relations, offering a more comprehensive yet abundant source of information for graph clustering.

Toward this end, in this paper, we propose **`DisenCluster`**, a novel **Disen**tangled graph representation learning framework for deep graph **Cluster**ing, which is capable of seeking a cluster structure that effectively spans across multiple relationships within a graph. In contrast to existing works that exploit entangled influential factors for graph clustering, our `DisenCluster` distinguishes various underlying latent factors within the graph data and explicitly fuses the corresponding node separation outcomes to achieve a consensus clustering result. In particular, we first leverage a disentangled graph convolutional layer to discern the factor responsible for the link between a given node and its neighbors, and accordingly extract pertinent features from the neighbors. Then, a contrastive loss is adopted for each component of the disentangled representation to make the node clustering-friendly. Hence, different clustering results could be individually generated through disentangled representations to investigate the underlying structure of each relation. We represent these clustering results as anchor graphs and seamlessly integrate all of them into a unified graph. Finally, a probabilistic cluster assignment matrix is learned by minimizing relaxed formulations of the graph cut problem, which is founded on the high-order connections within the constructed unified graph. To summarize, we make the following contributions:

- We propose to learn the disentangled representation for deep graph clustering. To the best of our knowledge, this is the first attempt to explicitly consider the underlying factors for graph clustering. The proposed mechanism can be potentially generalized to other unsupervised graph learning tasks.
- We propose a disentangled graph encoder to infer the clustering result for each underlying factor as anchor graphs and explicitly formulate the graph clustering problem as a continuous relaxation of the normalized graph cut problem founded on the high-order connections within the integrated unified graph.
- We perform thorough experiments across various benchmark datasets to evaluate the `DisenCluster`. The results demonstrate the effectiveness and strong interpretability of our proposed framework for the deep graph clustering task.

2 Related Work

2.1 Deep Graph Clustering

The fundamental idea of deep graph clustering is to leverage deep learning techniques, such as graph neural networks (GNNs) [16, 42, 57], to perform clustering on graph data. This approach aims to discover hidden community structures or groupings within a given graph, where nodes with similar characteristics are grouped together, facilitating data analysis and uncovering patterns and relationships within the data [3, 8, 23–25, 32, 39, 40, 61, 64]. On the one hand, numerous existing works adopt generative [3, 40] and adversarial [22, 32, 39] architectures to learn effective node representations. For example, DFCN [40] combines representations from AE and GAE, and incorporates target distribution generation and triplet self-supervision for enhanced cross-modality information utilization. Dink-Net [22] optimizes the cluster shrink loss and the cluster dilation loss in an adversarial manner for large scale graph clustering. On the other hand, many studies utilize the remarkable discriminative capabilities of contrastive learning [4, 35] to better serve clustering tasks. For instance, SCGC [24] enhances existing methods by implementing an innovative data augmentation technique and introducing a novel objective function for cross-view structural consistency. CCGC [61] leverages high-confidence clustering results to construct meaningful sample pairs for enhancing their discriminative capability and reliability. Additionally, there are some works on multi-view graph clustering that are highly relevant to our research [6, 9, 10, 14, 20, 31]. However, these approaches often fail to adequately disentangle the various underlying factors behind the graph data, thus ignoring the structure diversity of clustering results from diverse perspectives.

2.2 Disentangled Representation Learning

Disentangled representation learning seeks to decompose the representations to isolate the underlying explanatory factors in observed data [2] and has been widely applied across various domains [5, 19, 34, 46, 49, 51, 53, 60]. For example, β-VAE [13] adds the hyper-parameter β as the weight of the KL divergence for the Variational Autoencoders (VAEs) objective to balance the independence constraints and reconstruction accuracy. InfoGAN [5] decomposes the representation into noise and extra class code and estimates the mutual information between the class code and data for controllable image generation. MacridVAE [30] models user behavior data to study hierarchical user intentions via macro and micro representation disentanglement. DisenCite [53] learns the semantics of different sections in the paper via disentangled representation for context-specific citation generation. Recently, disentangled representation learning approaches have been increasingly explored in graph-structured scenarios [18, 29, 54, 63]. DisenGCN [29] introduces a neighborhood routing mechanism that enables the learning of disentangled node representations by partitioning each node's neighborhood into several disjoint subsets. FactorGCN [63] extracts block-wise interpretable features from the entire graph, facilitating graph-level classification tasks. DGCL [18] presents a contrastive learning-based self-supervised framework that learns disentangled representations without reliance on labeled data. Our work focuses on employing disentangled representation for deep graph clustering and the objective is to design a comprehensive framework capable of exploring and combining node separation results from diverse perspectives to effectively enhance the final clustering results.

3 METHODOLOGY

This section introduces the notations and problem definition with necessary background. The framework of `DisenCluster` is illustrated in Figure 2 and we present the details in the following.

3.1 Preliminaries

Notations. We denote a graph as a tuple $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with node set $\mathcal{V}$ and edge set $\mathcal{E} \in \mathcal{V} \times \mathcal{V}$, where $|\mathcal{V}| = N$ and $|\mathcal{E}| = M$. We use adjacency matrix $A \in \mathbb{R}^{N\times N}$ to characterize the structure information of $\mathcal{G}$, where entry $A(u, v) = 1$ if $(u, v) \in \mathcal{E}$ (otherwise entries of A are equal to 0). And $D = \mathrm{diag}(A1_N)$ is the degree matrix. The node feature matrix is represented as $X \in \mathbb{R}^{N\times d'}$, where each row $x_u \in \mathbb{R}^{d'}$ corresponds to the d'-dimensional feature vector of node u. In our work, we seek to develop a disentangled graph encoder, where the output node embedding $h_u^{(l)}$ can be factorized into K channels, i.e., $h_u^{(l)} = [h_{u,1}^{(l)}, h_{u,2}^{(l)}, \ldots, h_{u,K}^{(l)}]$, to facilitate the downstream deep graph clustering task.

Problem Definition. The goal of deep graph clustering is to partition the nodes within a graph into several disjoint groups in an unsupervised manner. We define a discrete cluster assignment matrix $R \in \{0, 1\}^{N\times C}$, where $R(u, j) = 1$ if $u \in \mathcal{S}_j$ and 0 otherwise. We denote $\mathcal{S}_j$ as the nodes belonging to the j-th cluster, which indicates the j-th column of R, namely, $R(j) = [R(1, j), \ldots, R(N, j)]^\top$.

Graph Cut and Normalised Cut. Graph clustering involves partitioning a graph into C disjoint groups, thereby nodes within intra-group are strongly connected than those between groups [43]. Thus, one of the most explicit and effective ways is to formulate the task as the Normalized Cut problem [38], expressed as:

$$\min_{\mathcal{S}_1,\ldots,\mathcal{S}_C} \sum_{j=1}^{C} \frac{\mathrm{cut}(\mathcal{S}_j, \bar{\mathcal{S}}_j)}{\mathrm{vol}(\mathcal{S}_j)}, \tag{1}$$

where $\bar{\mathcal{S}}_j$ denotes the node set not belong to cluster $\mathcal{S}_j$, namely can be, $\bar{\mathcal{S}}_j = \mathcal{V}\backslash\mathcal{S}_j$. Meanwhile, the numerator $\mathrm{cut}(\mathcal{S}_j, \bar{\mathcal{S}}_j) = \sum_{u\in\mathcal{S}_j, v\in\bar{\mathcal{S}}_j} A(u, v)$ means counting volume of edges that exist between different clusters, and the denominator $\mathrm{vol}(\mathcal{S}_j) = \sum_{u\in\mathcal{S}_j, v\in\mathcal{V}} A(u, v)$ means counting the edges between the nodes in a cluster and the rest of the graph. The normalized cut can be:

$$\begin{aligned} &\min_{\mathcal{S}_1,\ldots,\mathcal{S}_C} \sum_{j=1}^{C} \frac{\sum_{u\in\mathcal{S}_j, v\in\bar{\mathcal{S}}_j} A(u, v)}{\sum_{u\in\mathcal{S}_j, v\in\mathcal{V}} A(u, v)} \\ &= \max_{\mathcal{S}_1,\ldots,\mathcal{S}_C} \sum_{j=1}^{C} \frac{\sum_{u,v\in\mathcal{S}_j} A(u, v)}{\sum_{u\in\mathcal{S}_j, v\in\mathcal{V}} A(u, v)}, \end{aligned} \tag{2}$$

where the last equivalence follows from:

$$\sum_{u,v\in\mathcal{S}_j} A(u, v) + \sum_{u\in\mathcal{S}_j, v\in\bar{\mathcal{S}}_j} A(u, v) = \sum_{u\in\mathcal{S}_j, v\in\mathcal{V}} A(u, v). \tag{3}$$

3.2 Disentangled Graph Encoder

Given the graph $\mathcal{G}$ as input, we first assume that there are K underlying latent factors to be disentangled. Therefore, for each node $u \in \mathcal{V}$, we can project its feature x_u into different subspaces:

$$z_{u,k} = \frac{\sigma(W_k^\top x_u + b_k)}{\|\sigma(W_k^\top x_u + b_k)\|_2}, \tag{4}$$

where $W_k \in \mathbb{R}^{d'\times\frac{d}{K}}$ and $b_k \in \mathbb{R}^{\frac{d}{K}}$ are the weight of the k-th channel, $\sigma(\cdot)$ is a nonlinear activation function. We use the l_2 norm to ensure numerical stability and prevent neighbors with overly rich features.

For different factors of the representation, we define a collection of scoring matrices $\mathcal{A} = \{A_k|\forall k \in \{1, \ldots, K\}\}$ to represent finer-grained relationships between nodes. Within this collection, each entry $A_k(u, v)$ of the scoring matrix signifies the k-th relationship between node u and node v. We initialize them as $\mathcal{A}_k^0(u, v) = 1/K$ if $(u, v) \in \mathcal{E}$, which assumes that each relationship makes equal contributions at the initial stage and can be iteratively updated.

Within each channel k of node $u \in \mathcal{V}$, we aim to update independent component representation $z_{u,k}$. The main intuition is that we only use neighbor v connects with u due to factor k for the updating process. Specifically, at the iteration $i \in \{1, \ldots, I\}$, we assume the scoring entry $A_k^i(u, v)$ acting as an attention score, highlighting the primary driving factor behind a particular relationship between node u and v under factor k. We then perform message passing to update the corresponding components of node u's representation:

$$z_{u,k}^i = \frac{z_{u,k} + \sum_{v\in N(u)} A_k^{i-1}(u, v) \cdot z_{v,k}}{\|z_{u,k} + \sum_{v\in N(u)} A_k^{i-1}(u, v) \cdot z_{v,k}\|_2}, \tag{5}$$

where $z_{u,k}^i$ is the refined k-th component. Intuitively, nodes whose connections are driven by the same underlying factors tend to share similar representations within the corresponding components.

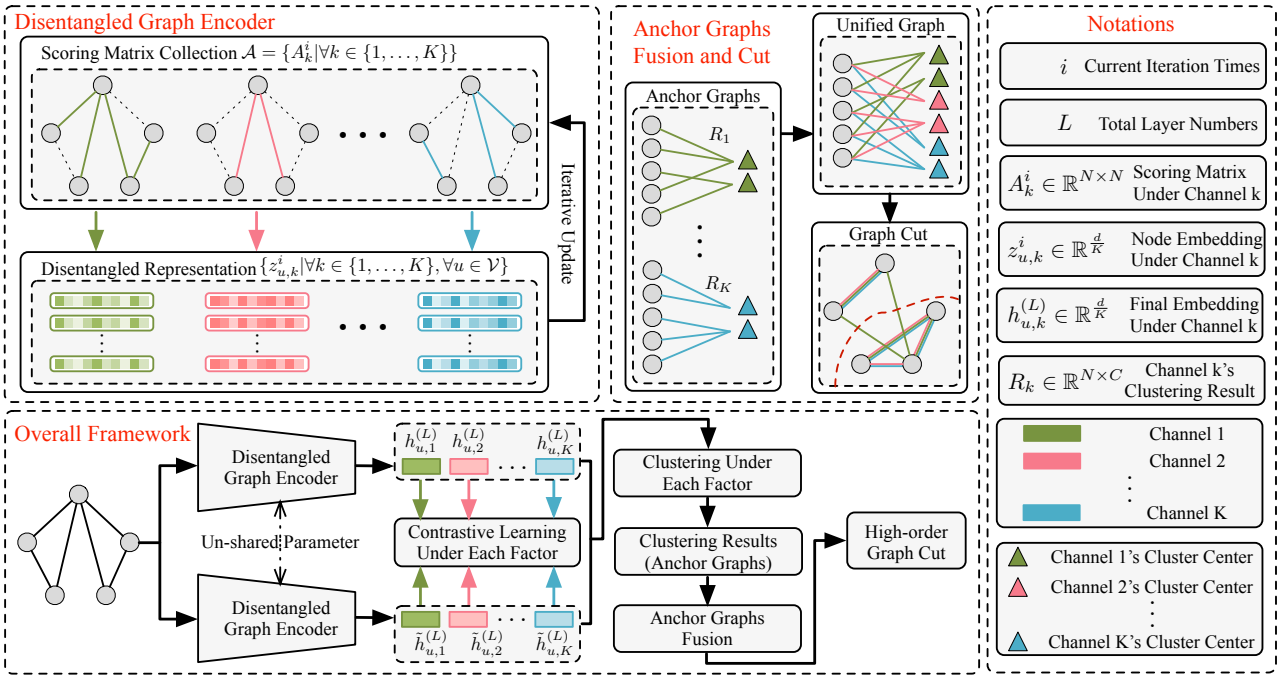


Figure 2: A schematic view of the `DisenCluster`. We first encode the two-view graph via the proposed Siamese disentangled graph encoders with unshared parameters. Then, we perform factor-wise contrastive learning to make each component of representation clustering-friendly. Following that, distinct clustering results are obtained under each factor and formulated as anchor graphs. Lastly, we integrate all anchor graphs into a unified graph and minimize the relaxed formulations of the graph cut problem. The process leverages the high-order connectivity within the unified anchor graph to ensure consistent cluster assignments across multiple factor channels, thereby enhancing the overall clustering performance.

Thus, we iteratively adjust the relationship strengths based on the updated representation, which is defined as follows:

$$A_k^i(u,v) = \frac{\exp({z_{v,k}}^\top z_{u,k}^i/\tau)}{\sum_{k'=1}^K \exp({z_{v,k'}}^\top z_{u,k'}^i/\tau)}, \tag{6}$$

where τ is the temperature parameter. After I iterations, we ultimately obtain the output of one disentangled graph convolutional layer, i.e., $h_{u,k}^{(1)} = z_{u,k}^I$. Notice that we can stack multiple disentangled graph convolutional layers to get $h_u^{(L)} = [h_{u,1}^{(L)}, h_{u,2}^{(L)}, \ldots, h_{u,K}^{(L)}]$, where L is the total stacking layers number. In this way, our model aggregates comprehensive factor-specific information and generates disentangled node representation.

3.3 Contrastive Graph Clustering

As effective representations benefit the clustering process, we introduce a factor-wise contrastive learning approach to enhance the discriminative qualities of the generated disentangled node representations. The idea is to create diverse graph sample views and define a self-supervised objective function by pulling together corresponding representations from the same sample while pushing away the different ones. To be specific, we construct different views of the graph with two Siamese disentangled graph encoders, where the parameters are un-shared between two sub-networks. We get node embedding matrix $H = \{h_1^{(L)}, \ldots, h_N^{(L)}\}^\top$ and $\tilde{H} = \{\tilde{h}_1^{(L)}, \ldots, \tilde{h}_N^{(L)}\}^\top$ and calculate the sample similarity matrix cross two views to formulate the contrastive loss,

$$L_{contra}^k(u) = -\log\Big(\frac{e^{sim(h_{u,k}^{(L)}, \tilde{h}_{u,k}^{(L)})}}{e^{sim(h_{u,k}^{(L)}, \tilde{h}_{u,k}^{(L)})} + \sum_{u \neq v} e^{sim(h_{u,k}^{(L)}, \tilde{h}_{v,k}^{(L)})}}\Big), \tag{7}$$

where $sim(\cdot)$ here denotes the similarity function. In practice, we set it as the cosine similarity function. Therefore, the contrastive loss can be calculated between the view pair under k-th channel. And the overall contrastive loss can be formulated as:

$$L_{contra} = \frac{1}{N} \sum_{u \in \mathcal{V}} \sum_{k=1}^K L_{contra}^k(u). \tag{8}$$

As such, we can minimize the contrastive objective function L_{contra} to keep the structural consistency across two views and enhance the discriminative capability of the learned disentangled representation, thereby further improving the clustering performance.

3.4 Anchor Graphs Generation and Fusion

Since each component of the disentangled representation stems from the different factors of one graph, the clustering result characterizes the inherent groupings or patterns presented within the corresponding semantic context. Thus, we implement a differentiable K-means clustering [56] based on each component of the

disentangled representation. Considering a node u and k-th component of its representation $h_{u,k}^{(L)}$, we assume μ_k^j as the center of cluster $\mathcal{S}_k^j$ and denote $R_k \in \mathbb{R}^{N\times C}$ as the cluster assignment matrix derived from the soft K-means. We employ a soft-min mechanism to assign nodes to cluster centers based on their distances.

$$R_k(u,j) = \frac{\exp(-\sigma\|h_{u,k}^{(L)} - \mu_k^j\|)}{\sum_{j'=1}^{C}\exp(-\sigma\|h_{u,k}^{(L)} - \mu_k^{j'}\|)}, \tag{9}$$

where the distance can also be negative cosine similarity, which shows strong performance. Meanwhile, the cluster centers can be refined iteratively following a process in typical K-means:

$$\mu_k^j = \frac{\sum_u R_k(u,j)h_{u,k}^{(L)}}{\sum_u R_k(u,j)}, \forall j = 1,\dots,C. \tag{10}$$

These iterated processes finally converge to a fixed point when μ_k^j remains consistent across consecutive updates. Note that the backward pass for this layer can be efficiently approximated without having to unfold the entire iteration sequence.

For each channel k of the graph, we formulate the clustering result R_k as an anchor graph which consists of two types of nodes, i.e., original graph nodes and anchors (cluster centers) respectively, and the edges connecting the anchor-node pairs reflect the weight to which node is assigned to the corresponding cluster. Since anchor sets obtained in different channels are not consistent when K-means clustering performs on each channel separately, we directly concatenate all the anchor sets as well as their connected edges to the graph nodes into a joint anchor graph $\mathcal{G}_R$ with adjacency matrix $A_R = R_1\|\dots\|R_K$, where the fused result $A_R \in \mathbb{R}^{N\times KC}$ and $\|$ means the matrix concatenation operation helping capture the cluster structure diversity of different channels.

3.5 High-order Graph Cut

Different from the Normalized Cut builds on first-order connectivity patterns (i.e., edges), we propose to cluster the nodes based on high-order connectivity of $\mathcal{G}_R$. The intuition is that when two nodes are consistently assigned to the same cluster across multiple channels, their influence on belonging to the same cluster in the final clustering result is strengthened. Formally, we introduce the high-order adjacency matrix $\hat{A} = A_R A_R^\top$, where each entry $\hat{A}(u,v)$ quantifies the frequency with which nodes u and v are assigned to the same cluster across various channels.

For now, we focus on high-order graph cut on the joint anchor graph $\mathcal{G}_R$, we rewrite Equation (2) as:

$$\begin{aligned}
&\max_{\mathcal{S}_1,\dots,\mathcal{S}_C}\sum_{j=1}^{C}\frac{\sum_{u,v\in\mathcal{S}_j}\hat{A}(u,v)}{\sum_{u\in\mathcal{S}_j,v\in\mathcal{V}}\hat{A}(u,v)}\\
&=\max_{R\in\{0,1\}^{N\times C}}\sum_{j=1}^{C}\frac{\sum_{u,v\in\mathcal{V}}\hat{A}(u,v)R(u,j)R(v,j)}{\sum_{u\in\mathcal{S}_j,v\in\mathcal{V}}R(u,j)\hat{A}(u,v)}\\
&=\max_{R\in\{0,1\}^{N\times C}}\sum_{j=1}^{C}\frac{R(j)^\top\hat{A}R(j)}{R(j)^\top\hat{D}R(j)}\\
&=\min_{R\in\{0,1\}^{N\times C}}-\mathrm{Tr}\Big(\frac{R^\top\hat{A}R}{R^\top\hat{D}R}\Big),
\end{aligned} \tag{11}$$

where $\hat{D} = \mathrm{diag}(\hat{A}1_N)$ is the degree matrix of $\hat{\mathcal{G}}$. The final cluster assignment matrix R can also generated by soft K-means with the whole representation of the node. We approximate the relaxed formulation of the high-order graph cut problem as follows:

$$L_{cut} = -\frac{1}{C}\cdot\mathrm{Tr}\Big(\frac{R^\top\hat{A}R}{R^\top\hat{D}R}\Big). \tag{12}$$

Furthermore, we introduce an auxiliary loss function to explicitly enforce hard cluster assignments,

$$L_{orth} = \frac{1}{\sqrt{C}-1}(\sqrt{C} - \frac{1}{N}\sum_{j=1}^{C}\|R(j)\|_F). \tag{13}$$

where $\|\cdot\|_F$ here indicates the Frobenius norm. And the orthogonality loss encourages the formation of well-balanced and distinguishable clusters, ensuring that nodes are primarily assigned to one cluster with high probability and less likely to be allocated to other clusters, which discourages further degenerate solutions.

3.6 Optimization

The proposed `DisenCluster` jointly optimizes three objectives, including contrastive loss L_{contra}, high-order graph cut loss L_{cut} and corresponding orthogonality loss L_{orth}. In summary, the objective of `DisenCluster` is formulated as:

$$L = L_{contra} + \alpha L_{cut} + \beta L_{orth}, \tag{14}$$

where α and β for the objective denote the trade-off parameters.

4 Experiments

4.1 Experimental Settings

Datasets. To evaluate the performance of our `DisenCluster`, in our experimental setting, we apply our model to six well-known and public accessible benchmark datasets following the previous works [25, 61], which includes Cora [36], Citeseer [7], Amazon Photo (AMAP) [37], Brazil AirTraffic (BAT) [24], Europe Air-Traffic (EAT) [24], and USA Air-Traffic (UAT) [24].

Baselines. To demonstrate the superiority of our `DisenCluster`, we adopt two groups of baselines for comparison, including (A) classical deep graph clustering methods, i.e., DAEGC [44], ARGA [32], SDCN [3] and DFCN [40], and (B) contrastive deep graph clustering methods, i.e., AGE [7], MVGRL [12], AutoSSL [15], AFGRL [17], GDCL [64], ProGCL [58], CCGC [61] and CDGC [62].

Evaluation Metrics. To ensure the comprehensive evaluation of the clustering methods, we adopt four benchmark metrics following [3] for evaluation: Accuracy (Acc), Normalized Mutual Information (NMI), Average Rand Index (ARI) and Macro F1-score (F1). The number of clusters C in the graph is set equal to the number of ground-truth classes in each dataset and the best map between cluster ID and class ID is constructed by the Kuhn-Munkres algorithm [33]. Larger values imply better clustering results. We perform ten repeated runs for each baseline method and report the mean and standard deviation of four metrics to reduce the impact of the random seed variation. The source code of `DisenCluster` is available at https://github.com/jamesyifan/DisenCluster.

Table 1: The comparison of clustering performance (%) across six benchmark datasets (mean±std). We highlight the best and second-best results among all methods in bold and underline, respectively.

Dataset	Metric	DAEGC	ARGA	SDCN	DFCN	AGE	MVGRL	AutoSSL	AGC-DRR	AFGRL	GDCL	ProGCL	CCGC	CDGC	DisenCluster
Cora	ACC	70.43±0.36	71.04±0.25	35.60±2.83	36.33±0.49	73.50±1.83	70.47±3.70	63.81±0.57	40.62±0.55	26.25±1.24	70.83±0.47	57.13±1.23	73.88±1.20	74.91±1.78	**78.33±0.93**
	NMI	52.89±0.69	51.06±0.52	14.28±1.91	19.36±0.87	57.58±1.42	55.57±1.54	47.62±0.45	18.74±0.73	12.36±1.54	56.30±0.36	41.02±1.34	56.45±1.04	58.16±0.83	**59.86±0.76**
	ARI	49.63±0.43	47.71±0.33	07.78±3.24	04.67±2.10	50.10±2.14	48.70±3.94	38.92±0.77	14.80±1.64	14.32±1.87	48.05±0.72	30.71±2.70	52.51±1.89	53.82±2.25	**57.81±1.77**
	F1	68.27±0.57	69.27±0.39	24.37±1.04	26.16±0.50	69.28±1.59	67.15±1.86	56.42±0.21	31.23±0.57	30.20±1.15	52.88±0.97	45.68±1.29	70.98±2.79	73.33±1.86	**76.78±1.72**
Citeseer	ACC	64.54±1.39	61.07±0.49	65.96±0.31	69.50±0.20	69.73±0.24	62.83±1.59	66.76±0.67	68.32±1.83	31.45±0.54	66.39±0.65	65.92±0.80	69.84±0.94	70.12±0.36	**72.28±0.15**
	NMI	36.41±0.86	34.40±0.71	38.71±0.32	43.90±0.20	44.93±0.53	40.69±0.93	40.67±0.84	43.28±1.41	15.17±0.47	39.52±0.38	39.59±0.39	44.33±0.79	43.56±0.35	**45.38±0.27**
	ARI	37.78±1.24	34.32±0.70	40.17±0.43	45.50±0.30	45.31±0.41	44.85±0.69	38.73±0.55	45.34±2.33	14.32±0.78	41.07±0.96	36.16±1.11	45.68±1.80	44.85±0.69	**48.65±0.31**
	F1	62.20±1.32	58.23±0.31	63.62±0.24	64.30±0.20	64.45±0.27	59.54±2.17	58.22±0.68	64.82±1.60	30.20±0.71	61.12±0.70	57.89±1.98	62.71±2.06	65.01±0.39	**66.42±0.61**
AMAP	ACC	75.96±0.23	69.28±2.30	53.44±0.81	76.82±0.23	75.98±0.68	41.07±3.12	54.55±0.97	76.81±1.45	75.51±0.77	43.75±0.78	51.53±0.38	77.25±0.41	77.24±0.87	**78.26±0.64**
	NMI	65.25±0.45	58.36±2.76	44.85±0.83	66.23±1.21	65.38±0.61	30.28±3.94	48.56±0.71	66.54±1.24	64.05±0.15	37.32±0.28	39.56±0.39	67.44±0.48	67.12±0.92	**67.79±0.90**
	ARI	58.12±0.24	44.18±4.41	31.21±1.23	58.28±0.74	55.89±1.34	18.77±2.34	26.87±0.34	60.15±1.56	54.45±0.48	21.57±0.51	34.18±0.89	57.99±0.66	58.14±0.82	**60.48±1.22**
	F1	69.87±0.54	64.30±1.95	50.66±1.49	71.25±0.31	71.74±0.93	32.88±5.50	54.47±0.83	71.03±0.64	69.99±0.34	38.37±0.29	31.97±0.44	72.18±0.57	**73.02±2.34**	72.22±0.63
BAT	ACC	52.67±0.00	67.86±0.80	53.05±4.63	55.73±0.06	56.68±0.76	37.56±0.32	42.43±0.47	47.79±0.02	50.92±0.44	45.42±0.54	55.73±0.79	75.04±1.78	75.50±0.87	**78.22±0.48**
	NMI	21.43±0.35	49.09±0.54	25.74±5.71	48.77±0.51	36.04±1.54	29.33±0.70	17.84±0.98	19.91±0.24	27.55±0.62	31.70±0.42	28.69±0.92	50.23±2.43	50.58±0.90	**54.51±1.07**
	ARI	18.18±0.29	42.02±1.21	21.04±4.97	37.76±0.23	26.59±1.83	13.45±0.03	13.11±0.81	14.59±0.13	21.89±0.74	19.33±0.57	21.84±1.34	46.95±3.09	47.45±1.53	**53.07±1.13**
	F1	52.23±0.03	67.02±1.15	46.45±5.90	50.90±0.12	55.07±0.80	29.64±0.49	34.84±0.15	42.33±0.51	46.53±0.57	39.94±0.57	56.08±0.89	74.90±1.80	75.40±0.88	**78.11±0.45**
EAT	ACC	36.89±0.15	52.13±0.00	39.07±1.51	49.37±0.19	47.26±0.32	32.88±0.71	31.33±0.52	37.37±0.11	37.42±1.24	33.46±0.18	43.36±0.87	57.19±0.66	57.22±0.73	**58.55±0.49**
	NMI	05.57±0.06	22.48±1.21	08.83±2.54	32.90±0.41	23.74±0.90	11.72±1.08	07.63±0.85	07.00±0.85	11.44±1.41	13.22±0.33	23.93±0.45	33.85±0.87	33.47±0.34	**33.95±0.63**
	ARI	05.03±0.08	17.29±0.50	06.31±1.95	23.25±0.18	16.57±0.46	04.68±1.30	02.13±0.67	04.88±0.91	06.57±1.73	04.31±0.29	15.03±0.98	27.71±0.41	26.21±0.81	**27.80±0.61**
	F1	34.72±0.16	52.75±0.07	33.42±3.10	42.95±0.04	45.54±0.40	25.35±0.75	21.82±0.98	35.20±0.17	30.53±1.47	25.02±0.21	42.54±0.45	57.09±0.94	57.53±0.67	**57.94±0.52**
UAT	ACC	52.29±0.49	49.31±0.15	52.25±1.91	33.61±0.09	52.37±0.42	44.16±1.38	42.52±0.64	42.64±0.31	41.50±0..25	48.70±0.06	45.38±0.58	56.34±1.11	55.31±2.42	**58.51±1.35**
	NMI	21.33±0.44	25.44±0.31	21.61±1.26	26.49±0.41	23.64±0.66	21.53±0.94	17.86±0.22	11.15±0.24	17.33±0.54	25.10±0.01	22.04±2.23	28.15±1.92	24.40±1.69	**29.03±1.19**
	ARI	20.50±0.51	16.57±0.31	21.63±1.49	11.87±0.23	20.39±0.70	17.12±1.46	13.13±0.71	09.50±0.25	13.62±0.57	21.76±0.01	14.74±1.99	25.52±2.09	22.14±1.67	**28.43±1.68**
	F1	50.33±0.64	50.26±0.16	45.59±3.54	25.79±0.29	50.15±0.73	39.44±2.19	34.94±0.87	35.18±0.32	36.52±0.89	45.69±0.08	39.30±1.82	55.24±1.69	52.77±2.61	**56.91±1.96**

4.2 Performance Comparison

We present the quantitative results of our proposed DisenCluster, compared with two groups of competitive deep graph clustering baselines, as reported in Table 1. And from the results, we can derive following key observations:

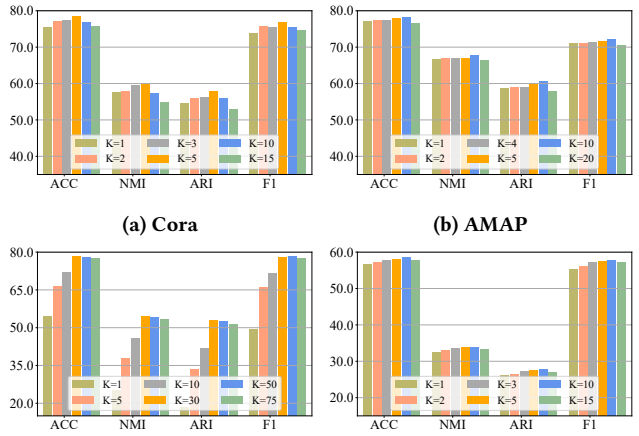
- Classicical deep graph clustering methods, which include some reconstructive and adversarial approaches, often perform notably worse than contrastive graph clustering approaches. This indicates that contrastive learning is an effective way to capture the supervision information for graph-structured data. By learning discriminative representations in a principled manner, contrastive learning better serves the clustering task.
- Contrastive graph clustering approaches achieve suboptimal performance compared with ours. This is primarily due to the intricate semantic nuances associated with different factors in real-world graphs. Traditional contrastive learning approaches, when applied to graph-structured data, are less effective in exploiting the intrinsic semantic information of the graph, resulting in consistently distinct clustering outcomes for the graph data.
- Overall, our proposed DisenCluster consistently outperforms other baselines on all six datasets and significantly outperforms the runner-up on many datasets. For instance, on the Cora dataset, we surpass the runner-up by 4.57%, 2.92%, 7.41%, and 4.70% on four evaluation metrics, which significantly demonstrates the effectiveness of our proposed model. We attribute the performance gain to two factors: (i) The effectiveness of capturing intricate semantic nuances for the graph. We encode the graph features into the disentangled representation, which notably captures the latent semantics of the graph. (ii) The distinct clustering results combination framework. We construct anchor graphs from clustering results and integrate them via high-order graph cut, which explicitly captures the diverse structure under different semantics and can be beneficial for graph clustering.

Table 2: Ablation studies on six datasets. Our full model achieves the best performance consistently.

Dataset	Metric	w/o Siam	w/o Contra	w/o Cut	w/o Orth	DisenCluster
Cora	ACC	77.36±0.68	70.67±1.77	76.39±0.45	76.91±0.48	**78.33±0.93**
	NMI	58.41±1.11	49.82±1.79	57.92±1.10	57.76±1.26	**59.86±0.76**
	ARI	56.66±1.07	46.86±2.19	54.73±0.69	55.85±0.87	**57.81±1.77**
	F1	75.93±1.10	68.19±2.51	73.05±1.62	75.83±1.14	**76.78±1.72**
Citeseer	ACC	72.06±0.43	66.19±1.76	71.24±0.54	71.98±0.57	**72.28±0.15**
	NMI	44.99±0.49	39.81±1.84	44.01±0.55	45.03±0.38	**45.38±0.27**
	ARI	48.16±0.59	39.45±1.43	47.15±0.62	48.33±0.38	**48.65±0.31**
	F1	66.10±1.05	60.32±2.06	66.07±0.49	66.11±0.69	**66.42±0.41**
AMAP	ACC	75.93±0.73	74.83±0.43	75.63±0.55	76.74±0.39	**78.26±0.64**
	NMI	63.17±1.38	64.96±0.69	64.37±0.85	65.52±0.78	**67.79±0.90**
	ARI	56.99±1.74	55.96±0.89	55.73±0.98	58.04±0.66	**60.48±1.22**
	F1	69.61±2.16	69.74±1.01	70.11±0.53	71.01±1.02	**72.22±0.63**
BAT	ACC	77.45±0.49	73.64±2.49	76.69±0.70	77.45±0.34	**78.22±0.48**
	NMI	52.88±0.79	48.19±2.46	51.24±1.17	53.52±0.52	**54.51±1.07**
	ARI	51.12±0.95	44.68±3.59	49.81±1.38	51.92±0.60	**53.07±1.13**
	F1	77.53±0.50	73.82±2.51	76.66±0.71	76.50±0.36	**78.11±0.45**
EAT	ACC	57.39±0.39	57.23±0.51	56.87±0.63	57.14±0.49	**58.55±0.49**
	NMI	33.49±0.62	31.47±1.79	32.08±1.39	33.14±0.62	**33.95±0.63**
	ARI	27.40±0.62	26.55±1.24	26.14±1.62	27.31±0.85	**27.80±0.61**
	F1	56.87±0.50	55.89±0.77	55.87±0.99	57.16±1.08	**57.94±0.52**
UAT	ACC	57.74±1.17	54.88±1.11	57.08±1.20	57.35±0.96	**58.51±1.35**
	NMI	27.03±1.56	24.84±1.03	25.95±1.37	27.32±1.05	**29.03±1.19**
	ARI	27.71±1.58	23.96±1.29	26.39±1.38	28.06±1.35	**28.43±1.68**
	F1	56.16±1.80	51.87±1.59	55.55±2.07	55.06±1.45	**56.91±1.96**

4.3 Ablation Study

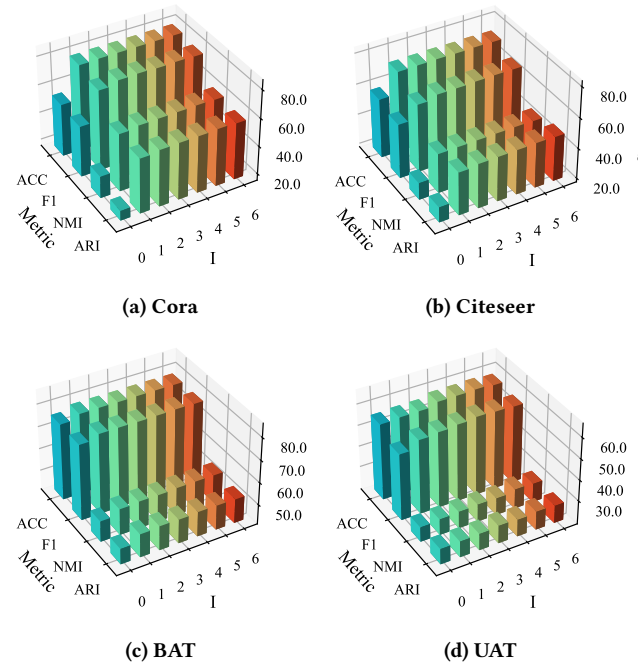
We conduct ablation studies on the key components of our proposed DisenCluster to gain deeper insights into its effectiveness. Specifically, we investigate a few variants to demonstrate how these components affect the model performance:



(a) Cora (b) AMAP

(c) BAT (d) EAT

Figure 3: Performance (%) w.r.t. different numbers of disentangled factors on four datasets.



(a) Cora (b) Citeseer

(c) BAT (d) UAT

Figure 4: Performance (%) w.r.t. different iteration times on four datasets.

- **w/o Siam** (without Parameter Un-shared Siamese Encoders): To simplify the graph augmentation process, we employ the parameter un-shared Simases encoders as two graph views. To verify its effectiveness, we introduce four other widely used graph data augmentations, including edge addition [59], edge dropping [59], graph diffusion [12], and feature masking [64] as variants, where the edge dropping/edge addition/graph teleportation/feature masking rate are all set as 20%. We report the best performance of these data augmentation methods.
- **w/o Contra** (without Contrastive Graph Clustering): We remove the contrastive graph clustering component and solely employ high-order graph cut loss and the corresponding orthogonality loss as the objective function ($L = L_{cut} + \beta L_{orth}$).
- **w/o Cut** (without High-order Graph Cut): We remove the high-order graph cut component and leverage K-means on the learned disentangled graph representation to obtain the final clustering results, and use both contrastive loss and orthogonality loss as the loss function ($L = L_{contra} + \beta L_{orth}$).
- **w/o Orth** (without Orthogonality Loss): We remove the orthogonality regularization loss and use both contrastive loss and high-order graph cut loss as the loss function ($L = L_{contra} + \alpha L_{cut}$).

The result of `DisenCluster` and its variants are shown in Table 2, from which we can draw the following key observations:

- Compared with conducting different node-views by parameter un-shared Simases encoders, the clustering performance is dropped by common graph augmentation strategies (w/o Siam). This indicates that augmentation strategies without careful design may lead to semantic drift [17]. Moreover, it demonstrates the effectiveness of our proposed parameter un-shared encoders.
- Without the contrastive graph clustering component (w/o Contra), the performance fails to achieve satisfactory performance. This indicates that factor-wise graph contrastive learning plays a key role in learning clustering-friendly node representation.
- Removing either the high-order graph cut component (w/o Cut) or orthogonality regularization (w/o Orth) also hinders the model performance, but to a lesser extent compared to removing the contrastive graph clustering component (w/o Contra). This reveals that both components make essential contributions to boosting the clustering performance.

4.4 Parameter Sensitivity

We also examine the sensitivity of the proposed `DisenCluster` to various hyper-parameters. Specifically, we investigate the effect of varying numbers of disentangled factors and iteration times for the graph disentangled encoder.

4.4.1 Impact of Disentangled Factor Number. To investigate whether `DisenCluster` can benefit from disentangled representations, we evaluate the model's performance with varying numbers of latent factors K in a disentangled graph encoder across four representative datasets. Figure 3 summarizes the experimental results, from which we derive the following key observations:

- In the case where $K = 1$, the model degenerates into a graph clustering framework based on entangled representations with poor performance. This observation underscores the importance of modeling various aspects of semantic information to greatly facilitate the clustering performance.
- Increasing the number of disentangled factors K can substantially enhance the clustering performance. Notably, `DisenCluster` achieves the best performance at $K = 5$ for Cora, $K = 10$ for AMAP/EAT and 30 for BAT, which represents the optimal aspects of semantics for graph clustering.
- However, when the number of disentangled factors is too large (i.e., $K \geq 10$ for Cora, AMAP and EAT), the performance will

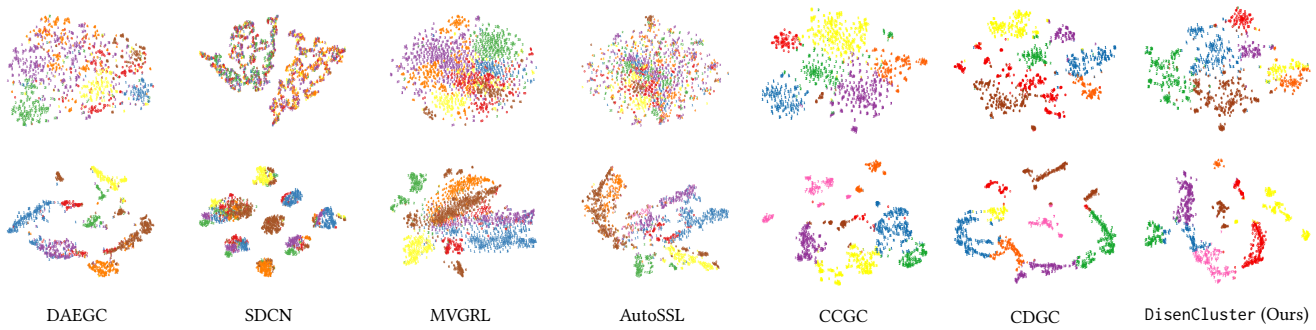
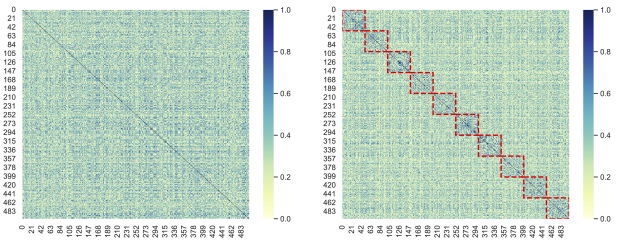


Figure 5: The t-SNE visualization on two datasets. The first and second rows represent Cora and AMAP, respectively.



(a) CDGC **(b) DisenCluster**

Figure 6: Representation correlation analysis. The absolute correlation values are computed between the elements of the 500-dimensional representations obtained through CDGC and our proposed DisenCluster with 10 channels, respectively, using the AMAP dataset.

degrade slowly. This may be attributed to the use of an overly complex semantic structure on the graph.

4.4.2 Impact of Iteration Times. To further examine how the number of iterations affects the model's performance, we maintain the optimal number of disentangled factors and explore different model variants with varying iteration times on Cora, Citeseer, BAT and UAT datasets. As shown in Figure 4, we set the iteration times for the disentangled graph encoder in the range of $[1, 6]$ and we have the following findings:

- Increasing the number of iterations typically results in improved performance until it stabilizes. For example, once the iteration times reach 3, the performance stabilizes, indicating that a small number of iterations is sufficient to achieve satisfactory results.
- Compared with the disentangled factor number, the model performance is not sensitive to the iteration times, especially on the BAT/UAT dataset. This can be attributed to the fact that training the model for multiple epochs can also drive convergence, even with fewer iteration times.

4.5 Visualization Analysis

In this part, we conduct two qualitative assessments to comprehensively investigate how our proposed DisenCluster facilitates graph clustering, including visualizations of the clustering result and the correlations between elements in learned representations.

4.5.1 Visualizations of Clustering Result. To unveil the inherent clustering structure, we plot the distributions of the learned representation on Cora and AMAP, compared with six baselines via t-SNE algorithm [41]. As depicted in Figure 5, our DisenCluster significantly improves cluster separability by fostering tighter aggregation of data points within individual clusters and expanding the separation margins between distinct clusters. This underscores the effectiveness of DisenCluster in learning highly discriminative feature representations, thereby facilitating more accurate and robust cluster assignments relative to existing approaches.

4.5.2 Visualizations of Representation Correlation. Besides the quantitative evaluation, we visualize the absolute correlation values between the elements of the learned representations. Figure 6 shows the correlation analysis of competitive baseline CDGC and our DisenCluster on the AMAP dataset. Although CDGC performs well on downstream tasks, its latent features are hidden entangled. By contrast, the correlation of DisenCluster reveals ten diagonal blocks, indicating that the highly independent representations across different factors. This demonstrates that disentangled representations make a significant contribution in capturing diverse semantic components and achieve a better performance in the graph clustering task.

5 Conclusion

In this paper, we propose a disentangled graph representation approach termed DisenCluster to improve the discriminative capability of nodes for deep graph clustering. Specifically, we propose a disentangled graph encoder to iteratively discerns the disentangled relationships between nodes, which are governed by different latent factors, and accordingly extracts the node representation from the corresponding channel. Then we propose a factor-wise contrastive loss to ensure the learned representations are clustering-friendly. Hence clustering results from different channels are formulated as anchor graphs and fused into one unified graph. Furthermore, we design a high-order graph cut to explore the cluster structure that effectively spans across multiple relationships within a graph. Extensive experiments on six benchmark graph clustering datasets demonstrate the efficacy of our DisenCluster. In future work, we will explore the application of our DisenCluster to a broader range of realistic graph representation-based downstream tasks, including graph classification and anomaly node detection.

Acknowledgments

This paper is partially supported by the National Key Research and Development Program of China with Grant No. 2023YFC3341203, the Fundamental Research Funds for the Central Universities in UIBE (Grant No. 23QN02), the National Natural Science Foundation of China under Grant 62276002 and 62306014, the Postdoctoral Fellowship Program (Grade A) of CPSF under Grant BX20250376, the Fundamental Research Funds for the Central Universities under Grant 1082204112K97, the Sichuan Science and Technology Program under Grant 2025ZNSFSC1506, and the Sichuan University Interdisciplinary Innovation Fund 1082204112J74.

References

[1] Lars Backstrom and Jure Leskovec. 2011. Supervised random walks: predicting and recommending links in social networks. In *Proceedings of the International ACM Conference on Web Search & Data Mining*. 635–644.

[2] Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 8 (2013), 1798–1828.

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

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

[5] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2016. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Proceedings of the Conference on Neural Information Processing Systems*. 2172–2180.

[6] Jiafeng Cheng, Qianqian Wang, Zhiqiang Tao, Deyan Xie, and Quanxue Gao. 2021. Multi-view attribute graph convolution networks for clustering. In *Proceedings of the International Joint Conference on Artificial Intelligence*. 2973–2979.

[7] Ganqu Cui, Jie Zhou, Cheng Yang, and Zhiyuan Liu. 2020. Adaptive graph encoder for attributed graph embedding. In *Proceedings of the International ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 976–985.

[8] Fnu Devvrit, Aditya Sinha, Inderjit Dhillon, and Prateek Jain. 2022. S3GC: Scalable self-supervised graph clustering. In *Proceedings of the Conference on Neural Information Processing Systems*. 3248–3261.

[9] Shaohua Fan, Xiao Wang, Chuan Shi, Emiao Lu, Ken Lin, and Bai Wang. 2020. One2multi graph autoencoder for multi-view graph clustering. In *Proceedings of the Web Conference*. 3070–3076.

[10] Uno Fang, Man Li, Jianxin Li, Longxiang Gao, Tao Jia, and Yanchun Zhang. 2023. A Comprehensive Survey on Multi-view Clustering. *IEEE Transactions on Knowledge and Data Engineering* (2023).

[11] Zhongkai Hao, Chengqiang Lu, Zhenya Huang, Hao Wang, Zheyuan Hu, Qi Liu, Enhong Chen, and Cheekong Lee. 2020. ASGN: An active semi-supervised graph neural network for molecular property prediction. In *Proceedings of the International ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 731–752.

[12] Kaveh Hassani and Amir Hosein Khasahmadi. 2020. Contrastive multi-view representation learning on graphs. In *International conference on machine learning*. PMLR, 4116–4126.

[13] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2016. beta-vae: Learning basic visual concepts with a constrained variational framework. In *Proceedings of the International Conference on Learning Representations*.

[14] Dong Huang, Chang-Dong Wang, and Jian-Huang Lai. 2023. Fast multi-view clustering via ensembles: Towards scalability, superiority, and simplicity. *IEEE Transactions on Knowledge and Data Engineering* (2023).

[15] Wei Jin, Xiaorui Liu, Xiangyu Zhao, Yao Ma, Neil Shah, and Jiliang Tang. 2022. Automated self-supervised learning for graphs. *Proceedings of the International Conference on Learning Representations*.

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

[17] Namkyeong Lee, Junseok Lee, and Chanyoung Park. 2022. Augmentation-free self-supervised learning on graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 7372–7380.

[18] Haoyang Li, Xin Wang, Ziwei Zhang, Zehuan Yuan, Hang Li, and Wenwu Zhu. 2021. Disentangled contrastive learning on graphs. In *Proceedings of the Conference on Neural Information Processing Systems*. 21872–21884.

[19] Hourun Li, Yifan Wang, Zhiping Xiao, Jia Yang, Changling Zhou, Ming Zhang, and Wei Ju. 2025. DisCo: graph-based disentangled contrastive learning for cold-start cross-domain recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 12049–12057.

[20] Zhiping Lin, Zhao Kang, Lizong Zhang, and Ling Tian. 2021. Multi-view attributed graph clustering. *IEEE Transactions on Knowledge and Data Engineering* (2021).

[21] Kang Liu, Feng Xue, Dan Guo, Peijie Sun, Shengsheng Qian, and Richang Hong. 2023. Multimodal graph contrastive learning for multimedia-based recommendation. *IEEE Transactions on Multimedia* (2023).

[22] Yue Liu, Ke Liang, Jun Xia, Sihang Zhou, Xihong Yang, Xinwang Liu, and Stan Z Li. 2023. Dink-net: Neural clustering on large graphs. In *Proceedings of the International Conference on Machine Learning*. 21794–21812.

[23] Yue Liu, Wenxuan Tu, Sihang Zhou, Xinwang Liu, Linxuan Song, Xihong Yang, and En Zhu. 2022. Deep graph clustering via dual correlation reduction. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 7603–7611.

[24] Yue Liu, Xihong Yang, Sihang Zhou, Xinwang Liu, Siwei Wang, Ke Liang, Wenxuan Tu, and Liang Li. 2023. Simple contrastive graph clustering. *IEEE Transactions on Neural Networks and Learning Systems* (2023).

[25] Yue Liu, Xihong Yang, Sihang Zhou, Xinwang Liu, Zhen Wang, Ke Liang, Wenxuan Tu, Liang Li, Jingcan Duan, and Cancan Chen. 2023. Hard sample aware network for contrastive deep graph clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 8914–8922.

[26] Yue Liu, Shihao Zhu, Jun Xia, Yingwei Ma, Jian Ma, Wenliang Zhong, Xinwang Liu, Shengju Yu, and Kejun Zhang. 2024. End-to-end Learnable Clustering for Intent Learning in Recommendation. In *Proc. of NeurIPS*.

[27] Yue Liu, Shihao Zhu, Tianyuan Yang, Jian Ma, and Wenliang Zhong. 2024. Identify Then Recommend: Towards Unsupervised Group Recommendation. In *Proc. of NeurIPS*.

[28] Jielong Lu, Zhihao Wu, Luying Zhong, Zhaoliang Chen, Hong Zhao, and Shiping Wang. 2024. Generative Essential Graph Convolutional Network for Multi-View Semi-Supervised Classification. *IEEE Transactions on Multimedia* (2024).

[29] Jianxin Ma, Peng Cui, Kun Kuang, Xin Wang, and Wenwu Zhu. 2019. Disentangled graph convolutional networks. In *Proceedings of the International Conference on Machine Learning*. 4212–4221.

[30] Jianxin Ma, Chang Zhou, Peng Cui, Hongxia Yang, and Wenwu Zhu. 2019. Learning disentangled representations for recommendation. In *Proceedings of the Conference on Neural Information Processing Systems*. 5712–5723.

[31] Erlin Pan and Zhao Kang. 2021. Multi-view contrastive graph clustering. In *Proceedings of the Conference on Neural Information Processing Systems*. 2148–2159.

[32] 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 (2019), 2475–2487.

[33] Michael D Plummer and László Lovász. 1986. *Matching theory*. Elsevier.

[34] Yifang Qin, Yifan Wang, Fang Sun, Wei Ju, Xuyang Hou, Zhe Wang, Jia Cheng, Jun Lei, and Ming Zhang. 2023. DisenPOI: Disentangling Sequential and Geographical Influence for Point-of-Interest Recommendation. In *Proceedings of the International ACM Conference on Web Search & Data Mining*. 508–516.

[35] Tao Ren, Haodong Zhang, Yifan Wang, Wei Ju, Chengwu Liu, Fanchun Meng, Siyu Yi, and Xiao Luo. 2025. MHGC: Multi-scale hard sample mining for contrastive deep graph clustering. *Information Processing & Management* 62, 4 (2025), 104084.

[36] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. 2008. Collective classification in network data. *AI magazine* 29, 3 (2008), 93–93.

[37] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. 2018. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868* (2018).

[38] Jianbo Shi and Jitendra Malik. 2000. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 8 (2000), 888–905.

[39] Zhiqiang Tao, Hongfu Liu, Jun Li, Zhaowen Wang, and Yun Fu. 2019. Adversarial graph embedding for ensemble clustering. In *Proceedings of the International Joint Conference on Artificial Intelligence*.

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

[41] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

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

[43] Ulrike Von Luxburg. 2007. A tutorial on spectral clustering. *Statistics and computing* 17 (2007), 395–416.

[44] Chun Wang, Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, and Chengqi Zhang. 2019. Attributed graph clustering: A deep attentional embedding approach. *arXiv preprint arXiv:1906.06532* (2019).

[45] Fuyun Wang, Xingyu Gao, Zhenyu Chen, and Lei Lyu. 2023. Contrastive Multi-Level Graph Neural Networks for Session-based Recommendation. *IEEE Transactions on Multimedia* (2023).

[46] Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. 2020. Disentangled graph collaborative filtering. In *Proceedings of the International ACM SIGIR Conference on Research & Development in Information Retrieval*. 1001–1010.

[47] Yiming Wang, Dongxia Chang, Zhiqiang Fu, and Yao Zhao. 2021. Consistent multiple graph embedding for multi-view clustering. *IEEE transactions on multimedia* 25 (2021), 1008–1018.

[48] Yifan Wang, Hourun Li, Ling Yue, Zhiping Xiao, Jia Yang, Changling Zhou, Wei Ju, Ming Zhang, and Xiao Luo. 2025. DANCE: Dual Unbiased Expansion with Group-acquired Alignment for Out-of-distribution Graph Fairness Learning. In *Proceedings of the International Conference on Machine Learning*.

[49] Yifan Wang, Xiao Luo, Chong Chen, Xian-Sheng Hua, Ming Zhang, and Wei Ju. 2024. DisenSemi: Semi-supervised graph classification via disentangled representation learning. *IEEE Transactions on Neural Networks and Learning Systems* (2024).

[50] Yifan Wang, Yifang Qin, Yu Han, Mingyang Yin, Jingren Zhou, Hongxia Yang, and Ming Zhang. 2022. Ad-aug: Adversarial data augmentation for counterfactual recommendation. In *Joint European conference on machine learning and knowledge discovery in databases*. 474–490.

[51] Yifan Wang, Yifang Qin, Fang Sun, Bo Zhang, Xuyang Hou, Ke Hu, Jia Cheng, Jun Lei, and Ming Zhang. 2022. DisenCTR: Dynamic graph-based disentangled representation for click-through rate prediction. In *Proceedings of the International ACM SIGIR Conference on Research & Development in Information Retrieval*. 2314–2318.

[52] Yifan Wang, Jianhao Shen, Yiping Song, Sheng Wang, and Ming Zhang. 2022. HE-SNE: Heterogeneous event sequence-based streaming network embedding for dynamic behaviors. In *2022 International Joint Conference on Neural Networks (IJCNN)*. 1–8.

[53] Yifan Wang, Yiping Song, Shuai Li, Chaoran Cheng, Wei Ju, Ming Zhang, and Sheng Wang. 2022. DisenCite: Graph-Based Disentangled Representation Learning for Context-Specific Citation Generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 11449–11458.

[54] Yifan Wang, Suyao Tang, Yuntong Lei, Weiping Song, Sheng Wang, and Ming Zhang. 2020. Disenhan: Disentangled heterogeneous graph attention network for recommendation. In *Proceedings of the International Conference on Information and Knowledge Management*. 1605–1614.

[55] Yifan Wang, Yangzi Yang, Shuai Li, Yutao Xie, Zhiping Xiao, Ming Zhang, and Wei Ju. 2025. GMR-Rec: Graph mutual regularization learning for multi-domain recommendation. *Information Sciences* 703 (2025), 121946.

[56] Bryan Wilder, Eric Ewing, Bistra Dilkina, and Milind Tambe. 2019. End to end learning and optimization on graphs. In *Proceedings of the Conference on Neural Information Processing Systems*. 4674–4685.

[57] Zhihao Wu, Xincan Lin, Zhenghong Lin, Zhaoliang Chen, Yang Bai, and Shiping Wang. 2023. Interpretable graph convolutional network for multi-view semi-supervised learning. *IEEE Transactions on Multimedia* (2023).

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

[59] Wei Xia, Qianqian Wang, Quanxue Gao, Ming Yang, and Xinbo Gao. 2022. Self-consistent contrastive attributed graph clustering with pseudo-label prompt. *IEEE Transactions on Multimedia* (2022), 1–13.

[60] Mengyue Yang, Furui Liu, Zhitang Chen, Xinwei Shen, Jianye Hao, and Jun Wang. 2021. Causalvae: Disentangled representation learning via neural structural causal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9593–9602.

[61] Xihong Yang, Yue Liu, Sihang Zhou, Siwei Wang, Wenxuan Tu, Qun Zheng, Xinwang Liu, Liming Fang, and En Zhu. 2023. Cluster-guided contrastive graph clustering network. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 10834–10842.

[62] Xihong Yang, Erxue Min, Ke Liang, Yue Liu, Siwei Wang, Sihang Zhou, Huijun Wu, Xinwang Liu, and En Zhu. 2024. Graphlearner: Graph node clustering with fully learnable augmentation. In *Proceedings of the ACM International Conference on Multimedia*. 5517–5526.

[63] Yiding Yang, Zunlei Feng, Mingli Song, and Xinchao Wang. 2020. Factorizable graph convolutional networks. In *Proceedings of the Conference on Neural Information Processing Systems*. 20286–20296.

[64] Han Zhao, Xu Yang, Zhenru Wang, Erkun Yang, and Cheng Deng. 2021. Graph Debiased Contrastive Learning with Joint Representation Clustering. In *Proceedings of the International Joint Conference on Artificial Intelligence*. 3434–3440.