Expert Systems With Applications 307 (2026) 130921

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Multi-view contrastive deep subspace clustering for attributed graph

Yanglan Gan [a], Zhengtian Gu [a], Guangwei Xu [a], Guobing Zou [b,*]

[a] School of Computer Science and Technology, Donghua University, Shanghai, 201620, China
[b] School of Computer Engineering and Science, Shanghai University, Shanghai, 200444, China

ARTICLE INFO

Keywords:
Multi-view clustering
Deep subspace clustering
Contrastive learning
Multi-scale information consistency

ABSTRACT

Multi-view clustering aims to discover inherent data structure by leveraging complementary perspectives of graph data. Although deep subspace clustering methods have achieved impressive performance, they usually construct the self-expression matrix using only the final embedding. This approach may overlook multi-scale information embedded across network layers and consistency between subspace representations and clustering labels. To address these limitations, we propose MvCDSC, an effective contrastive deep subspace clustering framework for multi-view graph-structured data. MvCDSC integrates view-specific and shared graph autoencoders to capture view-specific intricacies while learning cross-view shared representations. Its key innovations include two aspects. First, a multi-scale consistency mechanism aligns self-expression matrices across shallow and deep layers through layer-wise constraints. This captures multi-scale information and enriches the construction of representations. Second, an improved contrastive learning strategy is applied directly to the self-expression coefficient matrices. By redefining positive, false-negative, and negative pairs using pseudo-labels, this strategy effectively bridges semantic gaps across multiple views. Extensive experimental results demonstrate that MvCDSC outperforms state-of-the-art methods in node clustering tasks. The source code of the proposed MvCDSC is available at https://github.com/DHUDBlab/MvCDSC.

1. Introduction

Graph-structured data delineates intricate relationships among entities through nodes, edges, and attributes, playing significant roles in various domains of data analysis, such as citation networks and drug-drug interactions (Gan et al., 2023). Node clustering segregates graph nodes into distinct groups, aiming to maximize intra-cluster similarity and minimize inter-cluster similarity (Wang et al., 2019a). Previous clustering methods primarily focus on single-view data. However, real-world objects are often represented in multiple forms. Compared to single-view data, multi-view data provides richer information that can further refine clustering results. Nonetheless, multi-view clustering faces challenges such as significant modality gaps, dimensional disparities in node features, and diverse relation types, complicating view integration and multi-view clustering.

To date, a variety of multi-view clustering (MVC) methods have been proposed (Fang et al., 2023). Graph-based MVC methods (Lin & Kang, 2021; Pan & Kang, 2021; Wang et al., 2019b) seek to construct a consensus similarity graph by fusing or weighted approximating multiple similarity graphs, and then utilize graph-cut algorithms to obtain the clustering assignments. Matrix factorization-based MVC methods (Liu et al., 2013) decompose the data representation of each view into a basis matrix and a coefficient matrix, then implement constraints to derive the final coefficient matrix. Collaborative training-based MVC methods (Xu et al., 2021) iteratively select a view as the reference to guide the learning of other views, achieving impressive clustering performance through mutual learning between views. Kernel-based methods (Sun et al., 2021) construct a set of base kernels and a consensus kernel to process non-linear data, thus avoiding pre-defined problems. Although these traditional shallow models have achieved significant success, they also encounter difficulties in revealing nonlinear data relations and processing high-dimensional data. Recently, deep model-based MVC methods have garnered significant attention attributed to their remarkable learning capabilities. Particularly, based on the assumption that a data point can be expressed as a linear combination of other points in the same cluster, deep subspace clustering (DSC) (Cheng et al., 2022; Lele et al., 2022; Xia et al., 2021; Zhu et al., 2019) inserts a self-expression layer between the encoder and decoder to construct an affinity matrix for spectral clustering. The linear combination is captured by the self-expression coefficient matrices. This self-expression property is the core idea of subspace clustering. Observing that only utilizing the representation of the deepest hidden encoder layer for clustering might waste useful information embedded in other layers, Wang et al. (2021a) consequently constructed self-expression layers at each encoder layer,

* Corresponding author.
E-mail addresses: ylgan@dhu.edu.cn (Y. Gan), 2222785@mail.dhu.edu.cn (Z. Gu), gwxu@dhu.edu.cn (G. Xu), gbzou@shu.edu.cn (G. Zou).

https://doi.org/10.1016/j.eswa.2025.130921
Received 18 March 2025; Received in revised form 20 October 2025; Accepted 18 December 2025
Available online 30 December 2025
0957-4174/© 2026 Elsevier Ltd. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

enriching the self-expression coefficient matrix with multi-scale information. While deep subspace clustering methods have achieved impressive performance, they often construct the self-expression matrix based only on the final embedding. This critical omission ignores the multi-scale information embedded in different layers, leading to potentially sparse or incomplete coefficient matrices, and disregards the necessary consistency between the learned subspace representations and the final clustering labels (ABUD-ALLAH, 2024).

To make multiple clustering assignments consistent across different views (Xu et al., 2021), contrastive learning (Chen et al., 2020; He et al., 2020) has recently been adopted to enhance multi-view consistency. In traditional contrastive learning, the representations of the same node in different views forms a positive pair, while different nodes form negative pairs (Liu et al., 2022; You et al., 2020). Through node embedding, positive pairs are pulled closer and negative pairs are pushed away in the embedding space. This strategy lead to some similar samples from the same category being incorrectly classified as negative samples, which is not conducive to the downstream clustering task (Wang et al., 2017). Xia et al. (2022) proposed self-consistent contrastive learning, which redefined positive and negative pairs by introducing pseudo labels. In two views, for a node, it forms positive pairs with nodes with the same pseudo label, while forming negative pairs with the remaining nodes. However, pulling closer the node representations in different views is actually difficult because of large heterogeneous gap and semantic information difference. To alleviate the heterogeneity and promote better alignment of node representations, a shared auto-encoder is used to map node representations from different views into a common latent space. Yet another problem remains unsolved. Considering the semantic consistency that comes with self-expression coefficient matrices of different views, self-expression coefficient matrices might be more suitable for contrastive learning. On the other hand, since each view offers unique perspectives on the object, the integration of multiple views can enrich the clustering process with complementary information. For multi-view data, the fusion of information obtained from each view is a way to leverage complementarity. Cao et al. (2015) proposed an HSIC term to promote the diversity of subspace representations from different views, in order to further enhance the effect of complementarity. Achieving a balance between consistency and complementarity is still a challenging task now.

To address the aforementioned challenges, we propose a multi-view contrastive deep subspace clustering method (MvCDSC) for attributed graph. MvCDSC consists of multiple independent graph auto-encoders and a shared graph auto-encoder. Each auto-encoder is incorporated a self-expression layer to derive the self-expression coefficient matrix. Notably, we introduce an effective contrastive learning strategy specifically for deep subspace clustering, which redefines positive, false negative, negative samples according to pseudo labels obtained from clustering to promote semantic consistency across self-expression coefficient matrices of different views. With the shared auto-encoder and contrastive learning, our approach facilitates effective view fusion. Furthermore, to take advantage of multi-scale information, we minimize the disparity between the view-specific self-expression coefficient matrices and the fused one. This strategy not only ensures consistency in the decision space but also enhances the mutual complementarity between matrices corresponding to both shallow and deep topologies.

In summary, the major contributions of this paper are summarized as follow:

- We propose MvCDSC, an effective contrastive deep subspace clustering framework designed for multi-view graph-structured data. The framework employs view-specific graph autoencoders along with a shared graph autoencoder to capture the intricacies of each view while exploring shared information across views.
- We propose a new contrastive learning strategy that redefines positive, false negative, and negative pairs based on pseudo labels. This strategy is compatible with the self-expression coefficient matrix learning, effectively addressing the problems of heterogeneous and semantic gaps of multiple views.
- We respectively construct the self-expression coefficient matrices corresponding to shallow and deep graph convolution layers and ensure multi-scale information consistency.
- Extensive experiments on graph-structured datasets demonstrate that our proposed method outperforms state-of-the-art single-view and multi-view clustering methods. This end-to-end framework is applicable to two kinds of multi-view graph-structured datasets and demonstrates robust generalization.

2. Related work

2.1. Contrastive learning

Attributed to the unsupervised learning capability, contrastive learning has gained great attention in computer vision field (Chen et al., 2020; He et al., 2020). It aims to maximize mutual information by bringing positive pairs closer while simultaneously pushing away negative pairs in the embedding space. Several studies have introduced contrastive learning into graph learning. For example, GRACE (Zhu et al., 2020) leverages graph corruption techniques at both structural and attribute levels to generate two distinct graph views. Subsequently, node-level contrastive learning and the InfoNCE loss are applied to both views. Similarly, GraphCL (You et al., 2020) conducts graph-level contrast by summarizing graph representation through the readout function. Moreover, mutual information maximization can be performed at different scales. For instance, DGI (Velickovic et al., 2019) maximizes mutual information between node representations and a summary vector that captures global information of the entire graph. In GMI (Peng et al., 2020), each node incorporates node features and adjacency information of k-hop neighbors as its sub-graph, then maximizes mutual information between its node representation and the sub-graph. GCL4SR (Zhang et al., 2022) is a graph contrastive learning method for sequential recommendation, leveraging a Weighted Item Transition Graph (WITG) to capture global item transition patterns. It enhances sequence representations by aligning local and global contexts through subgraph-based contrastive learning and multi-scale consistency objectives. AdaGCL (Jiang et al., 2023) is a self-supervised contrastive learning framework for recommendation systems, which introduces adaptive view generators (a graph generator and a denoising model)to dynamically create contrastive views. It addresses data sparsity and noise by generating task-aware augmentations and mitigating model collapse through adversarial training. UMCGL (Du et al., 2024) is a multi-view consensus graph learning framework that considers both original and generative graphs to balance consistency and diversity. SMGCL (Zhou et al., 2023) constructs (node, subgraph) data pairs and (node, node) data pairs for contrast. However, contrastive learning usually ignores the category of the node, leading to some true positive pairs being mistakenly considered as negative pairs. Consequently, SCAGC (Xia et al., 2022) introduces a two-layer fully connected network on the Siamese network to obtain clustering labels, then refine contrastive learning. Similarly, CAGC (Wang et al., 2017) selects k-nearest neighbors in the latent space as positive pairs for each node, with the remaining nodes serve as negative pairs. PGCL-DCL (Hu et al., 2024) learns discriminative multi-view features and satisfactory clustering result by pseudo-label guided CL and dual correlation learning. MFCGC (Yang et al., 2024) employs a pseudo-label selection method to construct reliable positive and negative pairs.

2.2. Deep multi-view clustering

Deep learning models boost the development of multi-view clustering, facilitating the modeling of inter-view relations and the learning of latent representations. Utilizing a heuristic metric of modularity, O2MAC (Fan et al., 2020) selects the most informative view among multiple views to perform encoding, decoding, and self-training clustering.

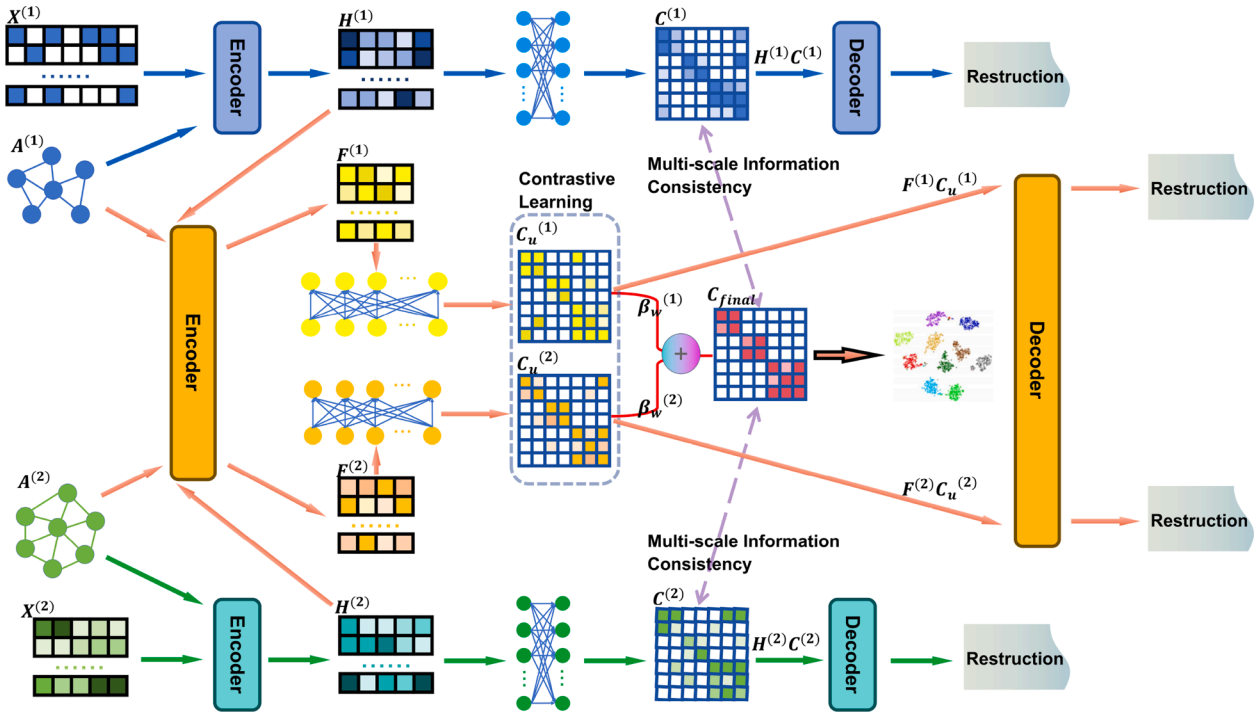

Fig. 1. The framework of the proposed method MvCDSC. MvCDSC consists of two main modules, including feature embedding module and self-expression matrix learning module. Multi-view graph data are first encoded by view-specific graph autoencoders to produce view-specific latent representations. These are then fed into a shared graph autoencoder to learn a common subspace. Self-expression layers are applied at multiple levels to generate coefficient matrices. A contrastive loss aligns cross-view self-expression matrices using pseudo-labels, while a multi-scale consistency loss regularizes shallow and deep matrices. The final fused coefficient matrix is used for spectral clustering.

Despite focusing on the most informative view, valuable information inherent in other views remains overlooked. SGCMC (Xia et al., 2021) introduces a new view construction method, and subsequently learns the view-consensus coefficient representation to facilitate spectral clustering, leveraging shared graph auto-encoder and geometric relationship similarity. A2AE (Sun et al., 2022) integrates the node representation of each view into a final representation using an attention mechanism. CMGEC (Wang et al., 2021b) and DFP-GNN (Xiao et al., 2023) integrate the adjacency matrix of each view into a consensus graph to incorporate complementary information. To tackle the issue of unbalanced feature dimensions, UMDL (Xu et al., 2023) constructs an overcomplete dictionary and utilizes a combination of atoms to transform the original data. However, these methods neglect to address the consistency between views, and the instability in model training caused by the attention mechanism. MvDSCN (Zhu et al., 2019) and SCMC (Lele et al., 2022) consider both consistency and complementarity. MGCCN (Liu et al., 2022) and DCMSC (Cheng et al., 2022) extend node-level contrastive learning to node representation, subspace representation, and high-order semantic representation.

3. Proposed method

MvCDSC is a new contrastive deep subspace clustering framework specifically designed for multi-view graph-structured data. As illustrated in Fig. 1, it consists of several main modules, including feature embedding module and self-expression matrix learning module. Specifically, feature embedding module introduces graph auto-encoder to map node attribute and adjacency matrix into a low-dimensional latent space, which is used for downstream node clustering task. Self-expression learning module consists of two parts, which introduce the improved contrastive learning and multi-scale information consistency minutely. The details will be elaborated in the following sections.

3.1. Notations

An undirected attribute graph is represented as $G = (N_e, E, X)$, where N_e is the nodes set, E is the edge set, and $X = \{x_1, x_2, \ldots, x_n\} \in \mathbb{R}^{n\times d}$ is the node attribute matrix. d is the dimension of node feature, n is the number of nodes, and $x_i \in R^d$ corresponds to the feature vector of node i. $A \in \mathbb{R}^{n\times n}$ is the adjacency matrix describing the connectivity among nodes. $A_{ij} = 1$ if $(v_i, v_j) \in E$, otherwise $A_{ij} = 0$.

For multi-view graph-structured data, it can be divided into two categories. One type is based on multi-attribute data, composed of multiple node attribute matrices and a common adjacency matrix, i.e. $X = \{X^{(v)} \in \mathbb{R}^{n\times d^{(v)}}\}_{v=1}^{V}$, $A \in \mathbb{R}^{n\times n}$. The other type is based on multi-layer data (Liu et al., 2022), composed of a node attribute matrix and multiple adjacency matrices, i.e. $X \in \mathbb{R}^{n\times d}$, $A = \{A^{(v)} \in \mathbb{R}^{n\times n}\}_{v=1}^{V}$. V represents the number of views, $d^{(v)}$ represents the dimension of node features in vth view. Multi-view graph clustering divides the n unlabeled nodes into k disjoint clusters by leveraging data collected from different sources or modalities, aiming to discover the underlying data structure.

3.2. Feature embedding module

In order to embed the node attributes and the relationship among nodes into a low-dimensional latent space, the feature embedding module introduces two types of graph auto-encoder, including view-specific and shared graph auto-encoder. View-specific graph auto-encoders are used separately for each view to unify the dimensions. The learned node representations and adjacency matrices are further fed into the shared graph auto-encoder. Then all nodes are embedded into a common latent space, facilitating subsequent fusion and contrastive learning. For the graph auto-encoder, similar to GATE (Salehi & Davulcu, 2019), we utilize the attention mechanism on GAE to discern the importance of

connections between nodes and their neighbors.

Encoder: Receiving the node attribute matrix X and adjacency matrix A as input, the encoder functions with L layers.

As the output of the lth layer, the representation of node i can be formulated as:

$$h_i^{(l)} = \sum_{j \in N_i} \alpha_{ij}^{(l)} \sigma(h_j^{(l-1)} W^{(l)}) \tag{1}$$

$$\alpha_{ij} = \frac{exp(e_{ij}^{(l)})}{\sum_{k \in N_i} exp(e_{ik}^{(l)})} \tag{2}$$

$$e_{ij}^{(l)} = sigmoid(\sigma(h_i^{(l-1)} W^{(l)}) v_s^{(l)} + \sigma(h_j^{(l-1)} W^{(l)}) v_r^{(l)}) \tag{3}$$

where $h_i^{(l)}$ is ith row of node representation matrix $H^{(l)}$, $W^{(l)} \in R^{d^{(l-1)} \times d^{(l)}}$, $v_s^{(l)}$, $v_r^{(l)} \in R^{d^{(l)}}$ are the trainable parameters of the lth encoder layer. N_i represents the neighbors of node i. $\alpha_{ij}^{(l)}$ represents the relevance of a neighboring node j to node i. $H^{(0)} = X$, $H^{(L)} \in R^{n \times d^{(L)}}$, $d^{(L)} < d$. $H^{(L)}$ is the output of the Lth encoder layer, which is chosen as the final representation of nodes. The encoding process reduces dimensionality and eliminates redundant information.

Decoder: We use a symmetric decoder to reconstruct the node attribute matrix X. Decoder takes node representation $\hat{H}^{(L)} = H^{(L)} C$ as input. The decoder utilizes previously calculated parameters $W^{(l)}$ and $\alpha_{ij}^{(l)}$ to reverse the encoding process, the node representation in the lth decoder layer can be formulated as:

$$\hat{h}_i^{(l-1)} = \sum_{j \in N_i} \alpha_{ij}^{(l)} \sigma(\hat{h}_j^{(l)} W^{(l)^T}) \tag{4}$$

where $\hat{h}_i^{(l)}$ is the ith row of $\hat{H}^{(L)}$. $\hat{X} = \hat{H}^{(0)}$ is the output of the last layer.

We respectively adopt two kinds of reconstructions to verify the quality of node representations. First, the node attribute reconstruction loss is defined as:

$$L_{AR} = ||X - \hat{X}||_F^2 \tag{5}$$

Meanwhile, to ensure that the representations of neighboring nodes are similar, the graph structure reconstruction loss is defined as:

$$L_{GR} = -\sum_{i=1}^{n} \sum_{j \in N_i} log\left(\frac{1}{1 + exp(-h_i h_j^T)}\right) \tag{6}$$

The total reconstruction loss in this module can be formulated as:

$$L_{re} = \sum_{v=1}^{2V} \left(||X^{(v)} - \hat{X}^{(v)}||_F^2 - \frac{1}{2} \sum_{i=1}^{n} \sum_{j \in N_i} log\left(\frac{1}{1 + e^{-h_i^{(v)} h_j^{(v)^T}}}\right)\right) \tag{7}$$

where $X^{(V+1)} = H^{(1)}$, …, $X^{(2V)} = H^{(V)}$, $H^{(V+1)} = F^{(1)}$, …, $H^{(2V)} = F^{(V)}$. λ_1 is a trade-off parameter. $H^{(v)}$ and $F^{(v)}$ respectively represent outputs of view-specific encoders and shared encoder for view v.

3.3. Self-expression matrix learning module

Before entering the decoder, $H^{(V)}$ will pass through the self-expression layer which is a fully connected layer devoid of activation functions and biases. The objective of this layer is to construct a linear combination expressed by other nodes. $C \in \mathbb{R}^{n \times n}$ is a self-expression coefficient matrix and a trainable weighted parameter of self-expression layer, C_{ij} measures the affinity between the ith node and the jth node. The self-expression layer is equipped for all graph auto-encoders. To optimize $C^{(v)}$, the self-expression loss is calculated as:

$$L_{se} = \frac{1}{2} \sum_{v=1}^{2V} ||H^{(v)} - H^{(v)} C^{(v)}||_F^2 + \mu ||C^{(v)}||_p \tag{8}$$

where $C^{(V+1)} = C_u{}^{(1)}$, …, $C^{(2V)} = C_u{}^{(V)}$. μ is always set to 10. $C^{(v)}$ and $C_u{}^{(v)}$ are self-expression coefficient matrices generated by view-specific auto-encoders and shared auto-encoder. To prevent the trivial solution $C^{(v)} = I$, we impose the constraint $C_{ii}^{(v)} = 0$. Additionally, $||C^{(v)}||_p$ serves as a sparsity penalty term, with p being set to 1.

3.3.1. Improved contrastive learning

While contrastive learning was initially developed for augmented views of single-modality data, recent advances have extended its principles to multi-view scenarios. Its goal is to learn representations across heterogeneous but semantically aligned views. In such frameworks, contrastive objectives are used to maximize mutual information between different views of the same instance, even when the views are not generated via augmentation. Our work builds upon this generalized view of contrastive learning, treating each graph view as a distinct modality of the same underlying structure. In traditional contrastive learning, positive pairs are constructed via data augmentation. In our multi-view setting, positive pairs are formed by linking different views of the same node. Although these views are not augmented copies, they share a common semantic identity-the node itself. Under the assumption that all views describe the same set of nodes, maximizing agreement between cross-view representations of the same node is equivalent to maximizing the mutual information across views.

Each row of the self-expression coefficient matrices represents a low-dimensional subspace representation of a node, encapsulating its affinities with other nodes. Despite being in different views, this innate semantic consistency is a huge advantage partnered with a shared auto-encoder to reduce heterogeneous gap. According to traditional contrastive learning, as shown in Fig. 2(a), the subspace representation of a node i serves as the anchor, while those of node i in other views are treated as positive samples, and the subspace representations of other nodes across views are considered negative samples. There are $v - 1$ positive pairs and $v(n - 1)$ negative pairs with the anchor. The objective is to minimize the distance between positive pairs and maximize the distance between negative pairs in the latent space. In general, nodes of the same class are expected to have similar linear combinations. This approach is not suitable when nodes of the same class as node i are mistakenly considered as negative samples and pushed away. Differently, for self-consistent contrastive learning (Xia et al., 2022), nodes with the same pseudo label as node i are also considered to be positive samples (Fig. 2(b)). However, due to the linear assumption of DSC, this inference is not necessarily valid for self-expression coefficient matrices, which prohibits nodes from self-representation. With the sparsity of graph-structured datasets, nodes belonging to the same class may exhibit dissimilar subspace representations. It is not appropriate to bring them closer. Self-consistent methods treats all same-label nodes as positives may force structurally inconsistent representations to align, violating the sparsity and self-expressiveness assumptions of DSC.

Therefore, as shown in Fig. 2(c), we propose a new contrastive learning strategy for DSC. Our strategy treats cross-view representations of the same node as positive pairs, while treating same-label but different-node representations as false negatives, allowing them to evolve naturally during training. This design avoids over-constraining the optimization and respects the inherent sparsity of subspace representations. Specifically, in the framework of a shared auto-encoder mapping a self-expression coefficient matrix across multiple views, nodes with the same index as node i are considered positive samples, those with the same pseudo label as node i are regarded as false negative samples, and the rest are treated as negative samples. These categories form positive, false negative and negative pairs with node i, with the objective of bringing positive pairs closer and distancing negative pairs, while do nothing with false negative pairs and let them update themselves during the learning process.

Here, spectral clustering provides pseudo labels to filtering out false negatives. Accordingly, similar to Xiao et al. (2023), we adopt a two-stage pre-training process. Initially, view-specific auto-encoders are pre-trained, followed by the shared auto-encoder. During formal training, these two types of auto-encoders employ distinct learning rates for back propagation, facilitating the preservation of clustering accuracy. As pseudo labels in two successive epochs might not be consistent, updating too quickly might lead to unstable model optimization. We take an interval T to update the pseudo labels, the value of T ranges from {5,

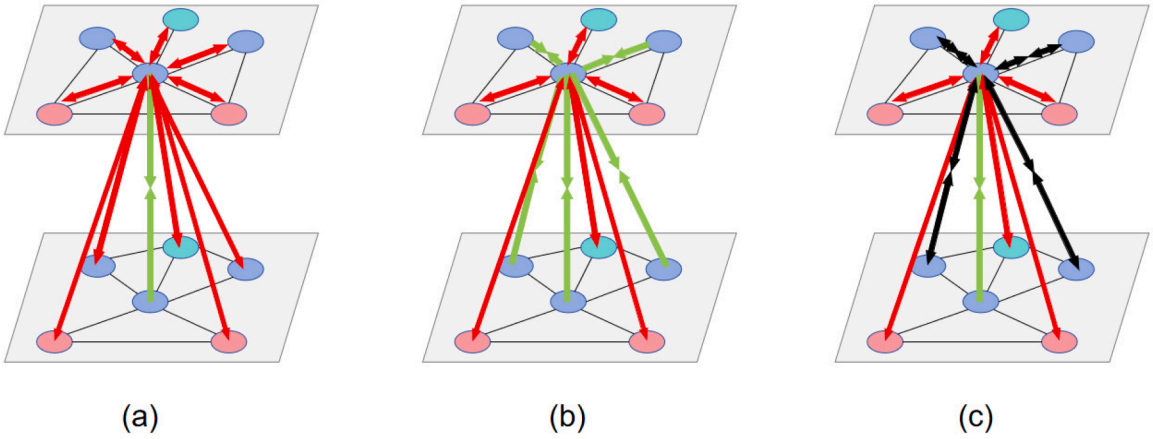

Fig. 2. Three types of contrastive learning. (a) traditional contrastive learning (b) self-consistent contrastive learning (c) the proposed self-expression contrastive learning. The red arrow means pushing away, the green arrow means pulling closer and the black arrow means neither pushing away nor pulling closer.

10, 15, 20}. We take two views as an example, the contrastive learning loss is calculated as:

$$f(C_i, C_j) = e^{sim(C_i^{(\alpha)}, C_j^{(\alpha)})/\tau} + e^{sim(C_i^{(\alpha)}, C_j^{(\beta)})/\tau} \tag{9}$$

$$L_{con} = -\frac{1}{2n} \sum_{\substack{\alpha=0 \\ \beta=1-\alpha}}^{1} \sum_{i=1}^{n} log \frac{e^{sim(C_i^{(\alpha)}, C_i^{(\beta)})/\tau}}{e^{sim(C_i^{(\alpha)}, C_i^{(\beta)})/\tau} + \sum_{j \in \nabla_i} f(C_i, C_j)} \tag{10}$$

where C_i represent the ith row of the self-expression coefficient matrix, ∇_i is the set of nodes that have different pseudo labels as i, τ denotes a temperature parameter, we set $\tau = 1$, α and β represents different views, $sim(C_i, C_j) = C_i^T C_j / ||C_i|| ||C_j||$.

Finally, we combine the self-expression coefficient matrices by a set of weight coefficients for complementarity:
$C_{final} = \sum_{v=1}^{V} \beta_w^{(v)} C_u^{(v)} \quad s.t. \sum_{v=1}^{V} \beta_w^{(v)} = 1, \beta_w^{(v)} \geq 0.$

Then the affinity matrix $\triangle$ can be calculated by $\triangle = \frac{1}{2}(|C_{final}| + |C_{final}^T|)$. Subsequently, spectral clustering can be applied to this matrix to derive the clustering results.

3.3.2. Multi-scale information consistency

Since the self-expression coefficient matrices derived from different layers represent distinct levels of feature representation, therefore a multi-scale consistency is introduced to address the critical problem: the potential inconsistency between the local structural information captured by shallow layers and the global semantic information captured by deep layers. By regularizing the alignment of these matrices, the model is forced to reconcile these two levels of information. This ensures that the learned representation is not only semantically meaningful but also structurally sound, which is crucial for a more stable and accurate final clustering result. To minimize the difference between self-expression coefficient matrices, the self-expression consistency loss L_{sc} is calculated as:

$$L_{sc} = \sum_{v=1}^{V} ||C_{final} - C^{(v)}||_F^2 \tag{11}$$

Besides, L_{sc} serves as a coordination mechanism for integrating multi-scale information within the model. It encompasses not only the final layer of the shared encoder but also the intermediate layers of view-specific encoders throughout the model's data flow. Specifically, the coefficients $C^{(v)}$ and $C_u^{(v)}$ are from distinct graph convolution layers. Shallow graph convolutions facilitate nodes in prioritizing neighboring nodes, whereas deep graph convolutions allow nodes to incorporate information from a broader spectrum of nodes. Through Eq. (11)], the self-expression coefficient matrices corresponding to shallow and deep topologies mutually complement each other. Consequently, the encoders are optimized towards achieving a refined self-expression coefficient matrix.

3.4. Overall loss function

To facilitate the end-to-end training, the total loss function of MvCDSC is formulated as:

$$L = \min_{W, v_{sr}, C} L_{re} + L_{se} + \lambda_1 L_{con} + \lambda_2 L_{sc} \tag{12}$$

where $W = \{W^{(1)}, W^{(2)}, \ldots, W^{(L)}\}$ and $v_{sr} = \{v_s^{(1)}, v_r^{(1)}, \ldots, v_s^{(L)}, v_r^{(L)}\}$ are training parameters of graph auto-encoder, C represents self-expression coefficient matrices, including those of the shared graph auto-encoder and the graph auto-encoder of each view. λ_1 and λ_2 are trade-off parameters. These modules are jointly trained in an end-to-end manner using the Adam algorithm (Kingma & Ba, 2014) with gradient clipping. The general procedure of MvCDSC are summarized in Algorithm 1.

4. Experiment

Having defined the architecture, the multi-scale consistency objective, and the contrastive loss function, this section will test the hypothesis that the fusion of multi-scale coefficient alignment and structure-guided contrastive learning improves clustering performance compared to state-of-the-art methods, particularly on complex attributed graph datasets.

4.1. Experiment setting

(1) Benchmark Dataset: We first evaluate the performance of our clustering approach on two types of graph-structured multi-view datasets, including multi-attribute and multi-layer datasets. For multi-attribute datasets, we choose three real-world datasets from diverse domains, including Cora, Citeseer, and Wiki (Yang et al., 2015). Cora and Citeseer represent citation networks, where nodes and edges signify publications and citations, respectively. The node attributes are presented as bag-of-words of keywords. Wiki represents a webpage network, where nodes and edges denote webpages and hyperlinks, respectively, and node attributes are described by TF-IDF weighted vectors. For the datasets initially containing only one attribute, we construct a second attribute through the Cartesian product of $X \times X^T$, where the original attribute and the transformed attribute share the same adjacency matrix. For multi-layer datasets, we choose two widely used datasets: ACM and IMDB (Sun et al., 2022), both with one attribute. ACM delineates a paper network with two types of relationships: co-paper and

Algorithm 1 The general procedure of MvCDSC.

Input: Node attribute matrices $X^{(1)}, \ldots, X^{(V)}$, adjacency matrices $A^{(1)}, \ldots, A^{(V)}$, hyper-parameters λ_1, λ_2, pre-training learning rate, formal training learning rate, maximum number of iterations $T_{pre1}, T_{pre2}, T_{formal}$, the dimension of graph auto-encoder, interval T, cluster number k, view number V.
Output: Clustering results

```
Initialize training parameters
for i = 1 : V do:
    for j = 1 : T_pre1 do :
        //pre-training view-specific graph auto-encoders
        Update W, v_sr, C of view-i graph auto-encoder using Adam
    end for
end for
for i = 1 : T_pre2 do:
    //pre-training shared graph auto-encoders
    Update W, v_sr, C of shared graph auto-encoder using Adam
end for
for i = 1 : T_formal do:
    Compute H^(1), ..., H^(V), C^(1), ..., C^(V) by solving Eq. (1), (2), (3), (7), (8)
    Compute F^(1), ..., F^(V), C_u^(1), ..., C_u^(V), C_final
    Update W, v_sr, C by minimizing Eq. (12) using Adam
    Run spectral clustering on △ = 1/2(|C_final| + |C_final^T|) to get the clustering results
    if i % T == 0:
        Update the pseudo labels used in Eq. (10)
end for
return clustering results
```

co-subject, and node attributes are represented as bag-of-words of keywords. IMDB represents a movie network comprising co-actor and co-director relationships, with node attributes described as bag-of-words of plots. To further validate the effectiveness of our MvCDSC on non-graph-structured and large-scale datasets, we assess its performance on three additional datasets, including Caltech101, Yale, and HHAR. Specifically, Caltech101 is an image dataset comprising 101 object categories and one background category. For this dataset, we extract six features and select 1984-dimensional and 512-dimensional features of all images using HOG and GIST as different views. The Yale dataset, also an image dataset, includes 165 grayscale photographs of 15 different individuals, each with 11 face images. We extract 4096-dimensional and 3304-dimensional features and construct a graph structure using K-nearest neighbors. The HHAR dataset contains 10,299 recordings from smart phones and smart watches, categorized into six human activities. For HHAR, we employ Euler transformation and K-nearest neighbors to construct a second view. Detailed information and statistical summaries of these datasets are provided in Table 1.

(2) Baseline Models: We compare the clustering performance of our proposed model with state-of-the-art clustering methods, including the following single-view and multi-view methods.

Single-view methods: K-means is a basis clustering algorithm to get k clusters through repeated interaction. LINE (Tang et al., 2015) is a classical graph embedding method which preserves first-order proximity and second-order proximity between nodes. GAE and VGAE (Kipf & Welling, 2016) combine graph convolution with auto-encoder and variational auto-encoder to learn node representation. MGAE (Wang et al., 2017) proposes a marginalization process on the node content and paired with a denoising graph auto-encoder to boost the interplay between graph structure and node content. GATE (Salehi & Davulcu, 2019) utilizes self-attention to measure the relevance between node and its neighbors, then aggregate node representation accordingly. DAEGC (Wang et al., 2019a) proposes a neighbor-aware end-to-end framework which integrates embedding learning and node clustering. SDCN (Bo et al., 2020) constructs a KNN graph and introduces structural information into deep clustering by combining the representation of auto-encoder and graph convolutional network. MSGA (Wang et al., 2021a) designs a multi-scale self-expression module to obtain a discriminative coefficient representation from each layer of the encoder. CAGC (Wang et al., 2017) put forward the region-level contrast and implements the instance-level contrastive learning on the node representation before and after the self-expression layer.

Multi-view methods: RMSC (Xia et al., 2014) is a multi-view spectral clustering method based on the Markov chain. PwMC (Nie et al., 2017) is a graph-based multi-view clustering method which employs weights to each view, and SwMC is its self-conducted weight version. PMNE (Liu et al., 2017) processes multilayer networks through network aggregation. GMC (Wang et al., 2019b) imposes a rank constraint on the unified graph matrix. MCGC (Pan & Kang, 2021) filter out the undesirable high-frequency noise via graph filtering and learn a consensus graph regularized by graph contrastive loss. O2MAC (Fan et al., 2020) selects the most informative view from multiple views by a heuristic metric modularity to take part in the reconstruction and clustering. MvAGC (Lin & Kang, 2021) integrates graph filter and high-order relations of graph structures in the graph-based multi-view clustering. CMGEC (Wang et al., 2021b) takes a simultaneous fusion of node representations and adjacency matrices and maintains the similarity of neighboring characteristics of each view in the latent space, the same is true for DFP-GNN (Xiao et al., 2023). MAGCN (Cheng et al., 2021) is designed with two-pathway encoders that map node embeddings and learn view-consistency information. MGCCN (Liu et al., 2022) introduces a generic and effective contrastive learning based GCN framework which uses pre-defined combination coefficients to aggregate the heterogeneous multi-view/multi-layer information. DMVGC (Liu et al., 2024) employs attention mechanisms in the decoders of different views and incorporates a collaborative self-training objective to align clustering outcomes. DuaLGR (Ling et al., 2023) enhances each view's graph using pseudo labels to mitigate the impact of non-homophilous edges and to extract high-level view-common information. DMCE (Zhao et al., 2023b) utilizes graph reconstruction and graph contrastive learning to integrate similarity graphs from different views. DGR (Zhao et al., 2023a) introduces a deep graph-based MVC method using residual GCN and orthogonal loss. MFCGC (Yang et al., 2024) devises a fair augmentation strategy for attributed graphs to guide representation learning, and also proposes a reliable pseudo-label selection method to enhance contrastive learning.

(3) Evaluation metrics: we evaluate the clustering performance using four widely used metrics: Accuracy (ACC), Normalized Mutual Information (NMI), Adjusted Rand Index (ARI). The higher values of these metrics mean better clustering quality.

(4) Parameter settings: For all baseline models, parameters are set according to the literatures to achieve optimal performance. In our model, the dimensions of each layer of the auto-encoder are explored within the range of {4096, 2048, 1024, 512, 256, 128, 64, 32} and keep consistent throughout pre-training and formal training. To achieve a balanced loss function and obtain optimal results, trade-off parameters λ_1, λ_2 are adjusted within the range of 10^{-3} to 10^3. Detailed parameter configurations are listed in Table 2. Here, (d_1, d_2) and (d_3, d_4) refer to the dimensions of view-specific encoders and shared encoder respectively, while lr_1 and lr_2 represent their learning rates. Since each view-specific graph autoencoder is initially pre-trained individually, setting a small learning rate lr_1 helps preserve the uniqueness of each view. The parameter T is set to 10, except for Wiki, Caltech101 and HHAR, where it is set to 20.

Table 1
Statistics of the datasets.

Datasets	Dataset Types	Feature Types	Nodes	Edges	Attributes	Classes
Cora	graph-structure (multi-attribute)	Bag of words of keywords, Cartesian transform	2708	5429	1433, 2708	7
Citeseer	graph-structure (multi-attribute)	Bag of words of keywords, Cartesian transform	3327	4732	3703, 3327	6
Wiki	graph-structure (multi-attribute)	TF-IDF, Cartesian transform	2405	17,981	4973, 2405	17
ACM	graph-structure (multi-layer)	Bag of words of keywords	3025	29281, 2210761	1830	3
IMDB	graph-structure (multi-layer)	Bag of words of keywords	4780	98010, 21018	1232	3
Caltech101	non-graph-structure (image)	Gabor, Wavelet Moments, CENHIST, HOG, GIST, LBP	9144	–	48, 40, 254, 1984, 512, 928	102
Yale	non-graph-structure (image)	intensity, LBP, Gabor	165	top-5 and top-2 nearest neighbors	3304, 4096, 6750	15
HHAR	non-graph-structure (record)	Sensor records, Euler transform	10,299	top-5 nearest neighbors	561, 561	6

Table 2
The parameter settings of MvCDSC in the formal training.

Dataset	(d_1,d_2)	(d_3,d_4)	lr_1	lr_2	λ_1	λ_2	β_w
Cora	[512,512],[1024,512]	[512,512]	0.00002	0.0002	1.0	0.003	(0.7,0.3)
Citeseer	[1024,512],[1024,512]	[512]	0.00002	0.0002	5.0	0.1	(0.7,0.3)
Wiki	[4096,512],[1024,512]	[512]	0.00002	0.0003	200	1.5	(0.7,0.3)
ACM	[4096,512],[1024,512]	[128]	0.00002	0.0002	5.0	10	(0.7,0.3)
IMDB	[4096,512],[1024,512]	[256,128]	0.00002	0.0001	1.0	0.001	(0.0,1.0)
Caltech101	[512,512],[512,512]	[512]	0.0005	0.005	10	0.5	(0.7,0.3)
Yale	[2048,512],[2048,512]	[512]	0.0001	0.0002	10	0.1	(0.7,0.3)
HHAR	[512,512],[512,512]	[256]	0.00005	0.0001	100	0.01	(0.7,0.3)

Table 3
The clustering results on multi-attribute datasets.

Methods	Input	Cora			Citeseer			Wiki		
		ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
K-means	best$X$	0.500	0.317	0.239	0.544	0.312	0.285	0.417	0.440	0.150
VGAE	A&best$X$	0.592	0.408	0.347	0.392	0.163	0.101	0.450	0.167	0.263
GATE	A&best$X$	0.658	0.527	0.451	0.616	0.401	0.381	0.465	0.428	0.316
MGAE	A&best$X$	0.684	0.511	0.448	0.661	0.412	0.414	0.514	0.485	0.350
DAEGC	A&best$X$	0.704	0.528	0.496	0.672	0.397	0.410	0.478	0.449	0.324
SDCN	A&best$X$	0.712	0.535	0.506	0.659	0.387	0.401	0.385	0.375	0.285
MSGA	A&best$X$	0.747	0.578	0.519	0.698	0.433	0.415	0.522	0.481	0.323
CAGC	A&best$X$	0.764	0.603	0.552	0.707	0.438	0.463	0.530	0.485	0.336
CMGEC	A&$X^{(1)}$&$X^{(2)}$	0.707	0.485	0.417	0.677	0.367	0.407	–	–	–
MAGCN	A&$X^{(1)}$&$X^{(2)}$	0.751	0.598	0.532	0.698	0.418	0.423	0.483	0.427	0.216
MGCCN	A&$X^{(1)}$&$X^{(2)}$	0.761	0.602	0.558	0.703	0.441	0.451	0.535	0.454	0.326
DMVGC	$A^{(1)}$&$A^{(2)}$&X	0.693	0.536	0.470	0.691	0.438	0.457	–	–	–
MFCGC	A&$X^{(1)}$&$X^{(2)}$	0.744	0.561	0.531	0.704	**0.447**	0.466	0.544	0.481	0.363
MvCDSC	A&$X^{(1)}$&$X^{(2)}$	**0.768**	**0.604**	**0.573**	**0.713**	0.441	**0.467**	**0.549**	**0.487**	**0.369**

4.2. Performance comparison

To estimate the clustering performance of our proposed MvCDSC, we run the aforementioned baseline models ten times and report the average score to avoid the randomness. For part of baseline models, we quote the clustering performance on four metrics from previous studies and mark it as '-' if not exists. The results are shown in Tables 3–6, where the bold values indicate the best performance. For single-view methods, we perform the methods on each view respectively and report the best results.

From the experimental results, we have the following observations:

- In most cases, the clustering methods that use both node feature and adjacency matrix tend to achieve better performance than those using only one of them. This underscores the significance of node features and adjacency graphs in clustering. Moreover, multi-view approaches usually outperform single-view methods due to their ability to leverage richer information.
- Compared with traditional shallow models, deep neural network based methods usually achieve better performance, owing to their capacity in extracting more useful information from node feature and adjacency matrix and combining them more efficiently.

Table 4
The clustering results on multi-layer datasets.

Methods	Input	ACM			IMDB		
		ACC	NMI	ARI	ACC	NMI	ARI
LINE-avg	$A^{(1)}\&A^{(2)}$	0.6479	0.3941	0.3432	0.4719	0.0063	-0.0090
PMNE	$A^{(1)}\&A^{(2)}$	0.6936	0.4648	0.4302	0.4958	0.0359	0.0366
RMSC	$A^{(1)}\&A^{(2)}$	0.6315	0.3973	0.3312	0.2702	0.0054	0.0018
PwMC	$A^{(1)}\&A^{(2)}$	0.4162	0.0332	0.0395	0.2453	0.0023	0.0017
SwMC	$A^{(1)}\&A^{(2)}$	0.3831	0.0838	0.0187	0.2671	0.0056	0.0004
GAE-avg	X&best$A$	0.6990	0.4771	0.4378	0.4442	0.0413	0.0491
DAEGC	X&best$A$	0.8909	0.6430	0.7046	0.3683	0.0055	0.0039
CAGC	X&best$A$	0.917	0.711	0.769	–	–	–
O2MAC	$X\&A^{(1)}\&A^{(2)}$	0.9042	0.6923	0.7394	0.4502	0.0421	0.0564
MvAGC	$X\&A^{(1)}\&A^{(2)}$	0.8975	0.6735	0.7212	0.5633	0.0371	0.0940
CMGEC	$X\&A^{(1)}\&A^{(2)}$	0.9089	0.6912	0.7232	0.4844	0.0514	0.0469
MGCCN	$X\&A^{(1)}\&A^{(2)}$	0.9167	0.7095	0.7688	0.5490	0.0567	0.1071
DuaLGR	$X\&A^{(1)}\&A^{(2)}$	0.9270	0.7320	0.7940	0.5421	0.0600	0.1348
MFCGC	$X\&A^{(1)}\&A^{(2)}$	**0.9276**	**0.7403**	**0.7983**	0.5429	**0.0893**	0.1093
MvCDSC	$X\&A^{(1)}\&A^{(2)}$	0.9210	0.7333	0.7800	**0.5936**	0.0687	**0.1336**

Table 5
The clustering results on image datasets.

Methods	Caltech101			Yale		
	ACC	NMI	ARI	ACC	NMI	ARI
K-means	0.1370	0.3040	0.0835	0.6097	0.6610	0.4220
GMC	0.1950	0.2379	0.0042	0.6182	0.6735	0.4336
O2MAC	0.1168	0.3089	0.0275	0.4945	0.5718	0.3336
MCGC	0.2430	0.3907	0.1249	0.7454	0.7494	0.4735
CMGEC	0.1898	0.4126	0.1807	0.4667	0.5413	0.2191
DFP-GNN	0.2025	0.4153	0.3367	0.3313	0.3910	0.2569
DMCE	0.2718	0.4682	0.3246	0.7261	0.7336	0.5243
DGR	0.2717	**0.4719**	0.3208	0.7667	0.7530	0.5295
MvCDSC	**0.3109**	0.4528	**0.4221**	**0.7694**	**0.7569**	**0.5349**

Table 6
The clustering results on large datasets.

Methods	HHAR		
	ACC	NMI	ARI
K-means	0.5736	0.6009	0.4639
O2MAC	0.7219	0.6336	0.5470
MAGCN	0.6695	0.7002	0.5783
MGCCN	0.7402	0.7094	0.5871
MFCGC	0.7606	0.7012	0.6152
MvCDSC	**0.8014**	**0.7472**	**0.6617**

- The proposed MvCDSC consistently outperforms most baseline methods across four evaluation metrics. On the Cora dataset, MvCDSC demonstrates improvements over the suboptimal method, CAGC, with increases of 0.4% in ACC, 0.1% in NMI, and 2.1% in ARI. On the IMDB dataset, compared to the second-best method MFCGC, MvCDSC shows enhancements of 5.07% in ACC, 4.74% in F1, and 2.43% in ARI. On the Caltech101 dataset, MvCDSC outperforms the strongest baseline DMCE, with improvements of 3.91% in ACC and a remarkable 9.75% in ARI.
- MvCDSC also performs well on non-graph-structured datasets, despite not being specifically designed for them. This robustness can be attributed to the power loss function and architecture, which maintain performance even when the autoencoder is replaced by MLP. This indicates that MvCDSC is versatile and applicable beyond graph-structured datasets.
- While the quality of pseudo-labels is important for the improved contrastive learning, MvCDSC effectively mitigates the impact of poor-quality pseudo-labels through the pre-training phase combined with well-designed model architecture and proper learning rate. This combination helps generate high-quality pseudo-labels. Notably, even on challenging datasets like Caltech101, where obtaining high-quality pseudo-labels is difficult, MvCDSC achieves better results. These results indicate the robustness of our improved contrastive learning strategy.

4.3. Ablation study

To validate the efficacy of various components in our model (12), we conduct an ablation study on the Cora, Citeseer, and IMDB datasets. The loss functions L_{re}, L_{se}, L_{con} and L_{sc} correspond to each functional module as previously described. All ablation experiments are conducted within the framework of pre-trained graph auto-encoders. Spectral clustering remains employed for C_{final}.

We compare four different training strategies for MvCDSC:

- Using only reconstruction loss and self-expression loss.
- Using reconstruction loss, self-expression loss and contrastive learning.
- Using reconstruction loss, self-expression loss and multi-scale consistency.
- The total loss function.

Table 7 illustrates the role of each functional module in enhancing the clustering performance of MvCDSC. Particularly, the two-stage pre-training and contrastive learning loss significantly contribute to this improvement. indicating L_{con} provides supplementary information for C_{final}. When L_{sc} term is removed, the clustering performance of MvCDSC decreases across all datasets. The results demonstrate that the multi-scale consistency may serve as a vital coordination mechanism to integrate multi-scale information within the model. On the IMDB dataset, with fusion weight coefficients [0,1], MvCDSC appears to degenerate into single-view clustering. However, the loss functions ensures its essence as a multi-view clustering model. Removal of L_{con} and L_{sc} compromises its performance, indicating the importance of adjusting and interacting self-expression coefficient matrices across views.

4.4. Parameter analysis

We examine the sensitivity of various parameters, including trade-off parameters (λ_1 and λ_2), dimensions and layers of the shared graph auto-encoder, and fusion weight coefficients of self-expression coefficient matrices. Additionally, we compare the proposed enhanced contrastive learning with previous contrastive learning strategies.

(1) The effect of trade-off parameters: We first analyze the influence of trade-off parameters. Take the Cora dataset as an example, we tune the tested parameter from 10^{-3} to 10^{3} and fix other parameters with the values in Table 2.

Table 7
Ablation study.

$L_{re}+L_{se}$	L_{con}	L_{sc}	Cora			Citeseer			IMDB		
			ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
✓			0.744	0.569	0.547	0.688	0.427	0.439	0.5443	0.0602	0.0894
✓			0.763	0.594	0.567	0.708	0.434	0.460	0.5863	0.0633	0.1238
✓	✓	✓	0.757	0.580	0.557	0.703	0.428	0.453	0.5790	0.0702	0.1246
✓	✓	✓	0.768	0.604	0.573	0.713	0.441	0.467	0.5936	0.0687	0.1336

Table 8
Comparison between different versions of contrastive learning on Cora and IMDB datasets.

Different versions of contrastive learning	Cora			IMDB		
	ACC	NMI	ARI	ACC	NMI	ARI
traditional contrastive learning	0.759	0.584	0.562	0.5857	0.0619	0.1221
self-consistent contrastive learning	0.761	0.584	0.565	0.5765	0.0632	0.1204
our improved contrastive learning	0.768	0.604	0.573	0.5936	0.0687	0.1336

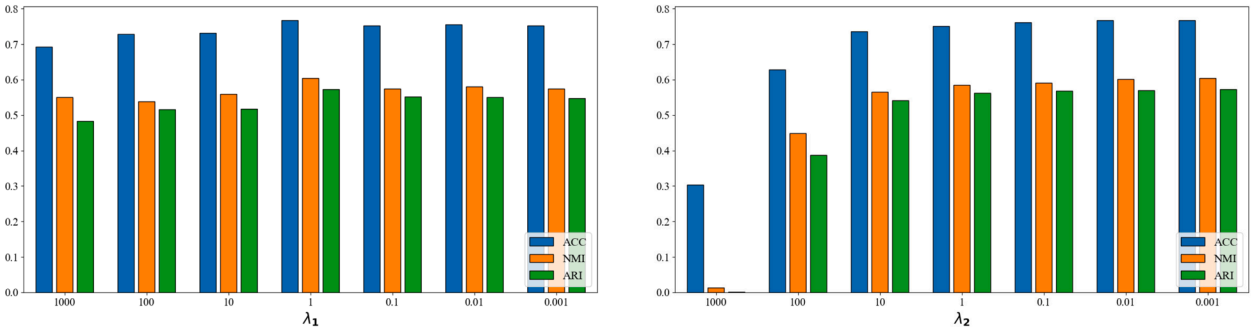

Fig. 3. The clustering performance (ACC, NMI and ARI) of individual trade-off parameter on Cora dataset.

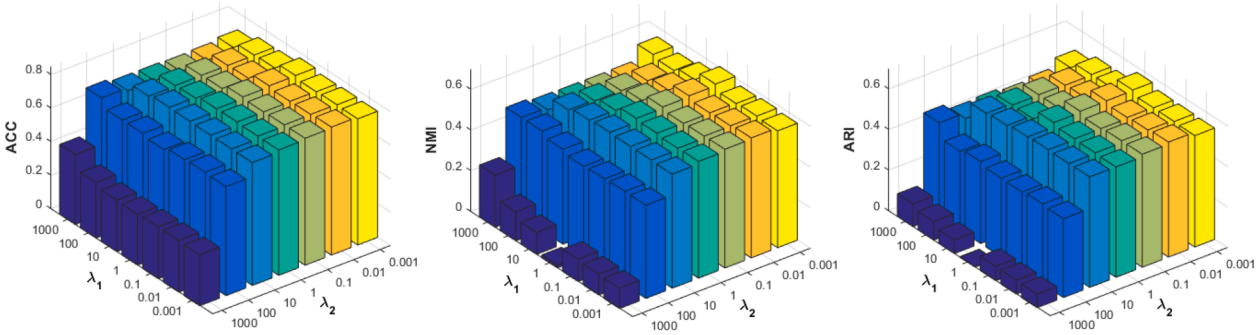

Fig. 4. The clustering performance (ACC, NMI and ARI) of λ_1 and λ_2 on Cora dataset.

Table 9
Computation time of several methods (seconds) on Cora and HHAR dataset.

Methods	Cora	HHAR
MAGCN	114.5	350.7
MGCCN	172.9	886.3
MFCGC	225.5	1393.0
SGCMC	300.4	1496.1
MvCDSC	220.3	1254.9

Fig. 3 shows the influence of individual trade-off parameters on the clustering performance. Notably, variations in λ_1 lead to minimal fluctuations in clustering metrics, indicating that the contrastive objective acts as a stable regularizer without overly dominating the overall training dynamics. As λ_2 gradually decrease, the clustering performance gradually converges to the optimum. When $\lambda_2 = 1000$, a decline is observed, suggesting that excessive emphasis on consistency may influence the fused self-expression coefficient matrix. This represents a failure case that excessive emphasis on cross-layer consistency overly constrains the view-specific encoders, suppressing their ability to capture unique, discriminative structural patterns. This leads to representation homogenization and degrades clustering quality. λ_2 relate to self-expression coefficient matrix optimization and information supplementation, which are critical for improving clustering metrics. The results indicate that contrastive learning can provide a reliable guarantee and only needs a small amount of information supplementation.

Subsequently, we analyze the combined impact of λ_1 and λ_2 on the Cora dataset. Both parameters ranging from 10^{-3} to 10^{3}. As illustrated in Fig. 4, the clustering performance exhibits improvement as λ_2 decreases from 10^3 to 10 and continues to rise marginally as λ_2 decreases to 10^{-3}. Notably, too high or small values of λ_1 are unsuitable for clustering. Optimal clustering performance are achieved when $\lambda_1 = 1$ and $\lambda_2 = 0.001$. The results demonstrate that L_{con} and L_{sc} can sufficiently explore multi-view consistency across multi-scale information and decision space, achieving good clustering results. This analysis provides

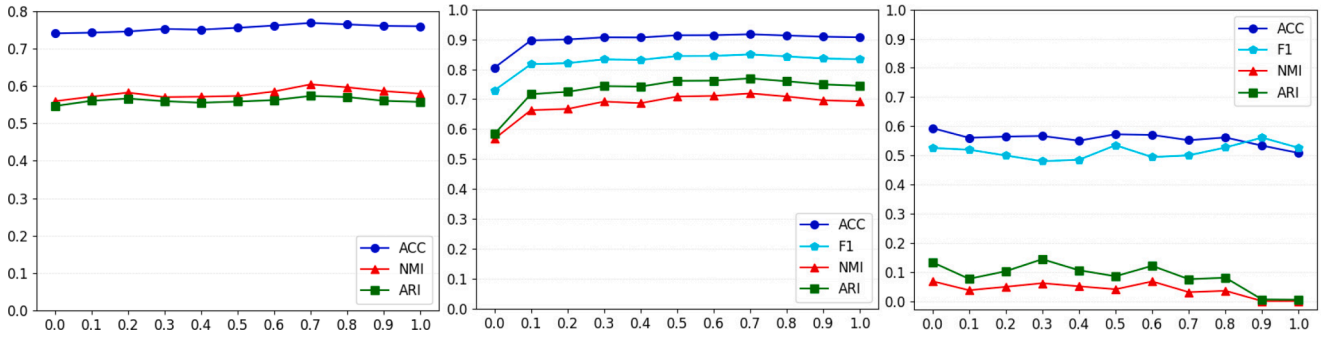

Fig. 5. The clustering performance (ACC, NMI and ARI) of weight coefficients on Cora, ACM, IMDB.

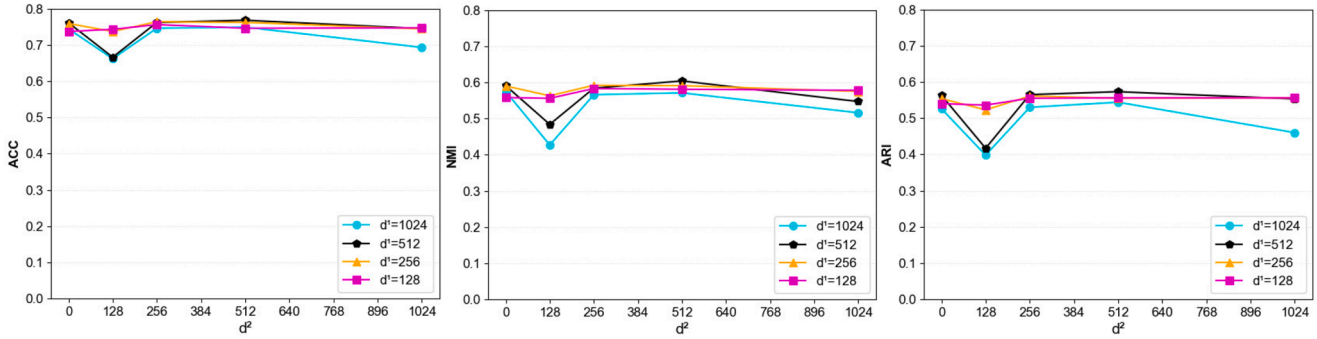

Fig. 6. Parameter sensitivity of d^1,d^2 on Cora dataset.

practical guidance for avoiding over-regularization in real-world deployment.

(2) The effect of weight coefficients: After obtaining the view-specific self-expression coefficient matrices, these matrices are combined using a set of weights. The impact of different combinations of weights on clustering performance is illustrated in Fig. 5. In this figure, the x-axis represents the weight of view 1, while the y-axis indicates clustering performance. Given that the sum of the weights for the two views is 1, the weight for the second view can be inferred accordingly. Overall, performance slightly degrades at both extremes, as the model effectively reduces to a single-view regime, losing the benefit of multi-view complementarity. For multi-attribute datasets, the second attribute is constructed via a Cartesian product, which is not the same as the original attribute. As for quality, the self-expression coefficient matrix corresponding to the original attribute should predominantly contribute to the fusion. For multilayer datasets, each view has a distinct adjacency graph that represents the different relationships between nodes, which influences the proximity of node representations in the embedding space. For the ACM dataset, the Paper-Author-Paper (PAP) relationship significantly enhances clustering performance compared to the Paper-Subject-Paper (PSP) relationship. Conversely, for the IMDB dataset, the Movie-Director-Movie (MDM) relationship is so influential that it another view is not needed for fusion. Based on the clustering performance of MvCDSC with different combinations of weights, we observe that MvCDSC exhibits remarkable robustness to weight assignment across views, empirically validating the parameter efficiency and operational feasibility of the fusion strategy.

Further, we evaluate the performance of MvCDSC with attention-based weights to compare the effect of different multi-scale fusion strategies. As shown in Supplementary Tables 1 and 2, comparative experiments at different learning rate reveal that attention-weighted fusion does not yield notable performance gains, but introduces significant computational overhead. Meanwhile, attention mechanisms introduce high sensitivity to learning rate tuning, requiring more hyperparameter optimization effort than our method.

(3) The effect of improved contrastive learning: To verify the effectiveness of the proposed contrastive learning strategy, we compare it with traditional contrastive learning and self-consistent contrastive learning. The difference among these types of contrastive learning lies in that traditional contrastive learning pushes away node pairs belonging to the same class as negative pairs, while self-consistent contrastive learning brings them closer. Our approach is to treat such node pairs as false negative pairs, neither pulling them closer nor pushing them away. Table 8 demonstrates that traditional strategies of pushing apart and pulling together exhibit varying performances across different datasets. In contrast, our neutral strategy consistently delivers superior results. Besides, it is important to note the impact of pseudo-label quality. Poor-quality pseudo-labels can cause our improved contrastive learning to regress to a form similar to traditional contrastive learning. Pseudo-labels of excessively high quality do not yield disproportionately large performance gains. To address this, we carefully design the model architecture, training epochs, and learning rate to ensure a decent quality of pseudo-labels at the beginning. Overall, regardless of pseudo-label quality, the improved contrastive learning improves the performance.

(4) The effect of dimensions and layers of shared graph auto-encoder: On different datasets, the dimensions and layers of shared graph auto-encoder often need to be adjusted. Take the Cora dataset as an example, we explore the dimensionality settings for each graph convolutional layer within the range of [128, 256, 512, 1024]. The number of graph convolutional layers in the shared graph auto-encoder are constrained to a maximum of two. As illustrated in Fig. 6, d^1 and d^2 represent the dimensions of the first and second layers of the shared graph auto-encoder, respectively. $d^2 = 0$ indicates a single-layer configuration. The four curves in the figure correspond to scenarios where d^1 is set to 1024, 512, 256, and 128, with d^2 varying accordingly. The results indicate that MvCDSC exhibits reasonable fluctuations in performance as the dimensions vary significantly. However, there is a notable decline in clustering performance when the difference in dimensions between the first and second layers is substantial. This suggests that more experimentation is required to optimize layer dimensions. Furthermore, the experiment demonstrates the robustness and rationality of our model structure.

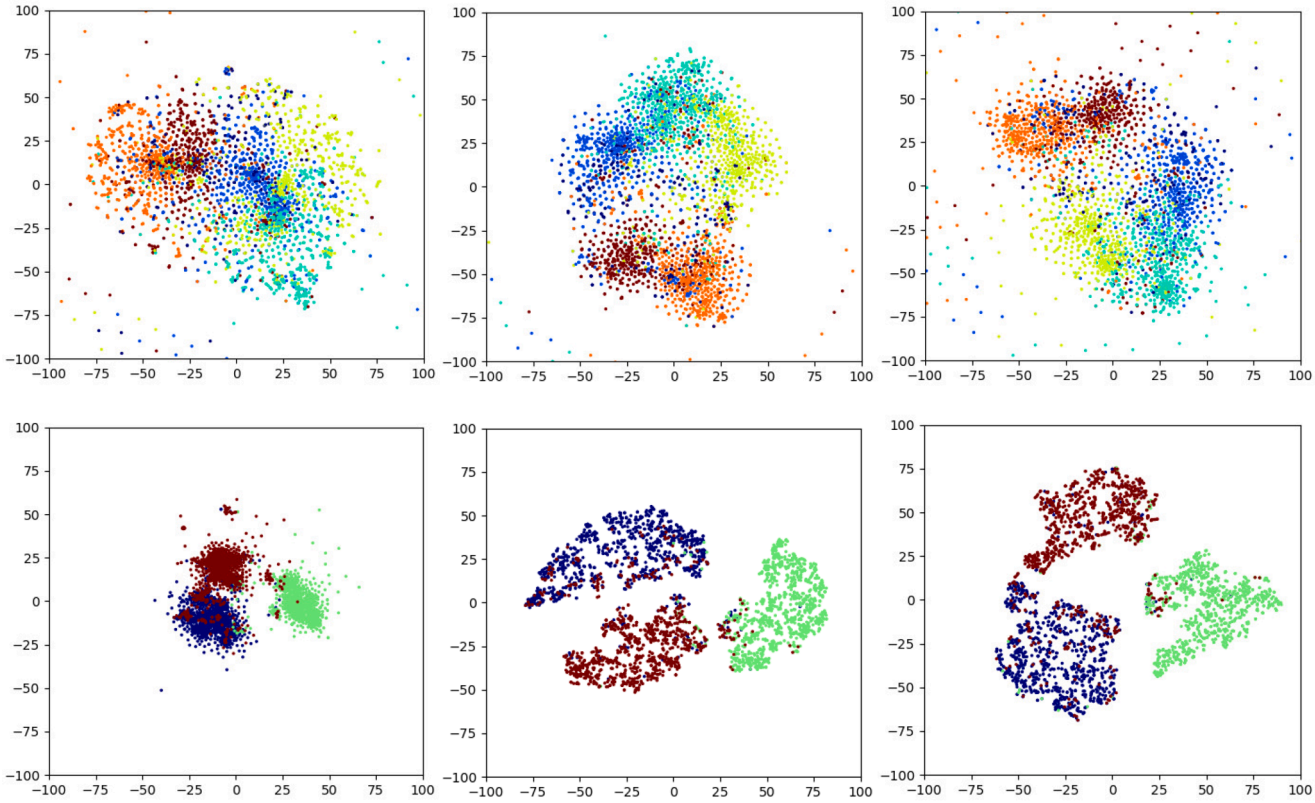

Fig. 7. The visualization of C_{final} on Citeseer dataset at epoch 0, epoch 149, epoch 205. The visualization of C_{final} on ACM dataset at epoch 0, epoch 13, epoch 35.

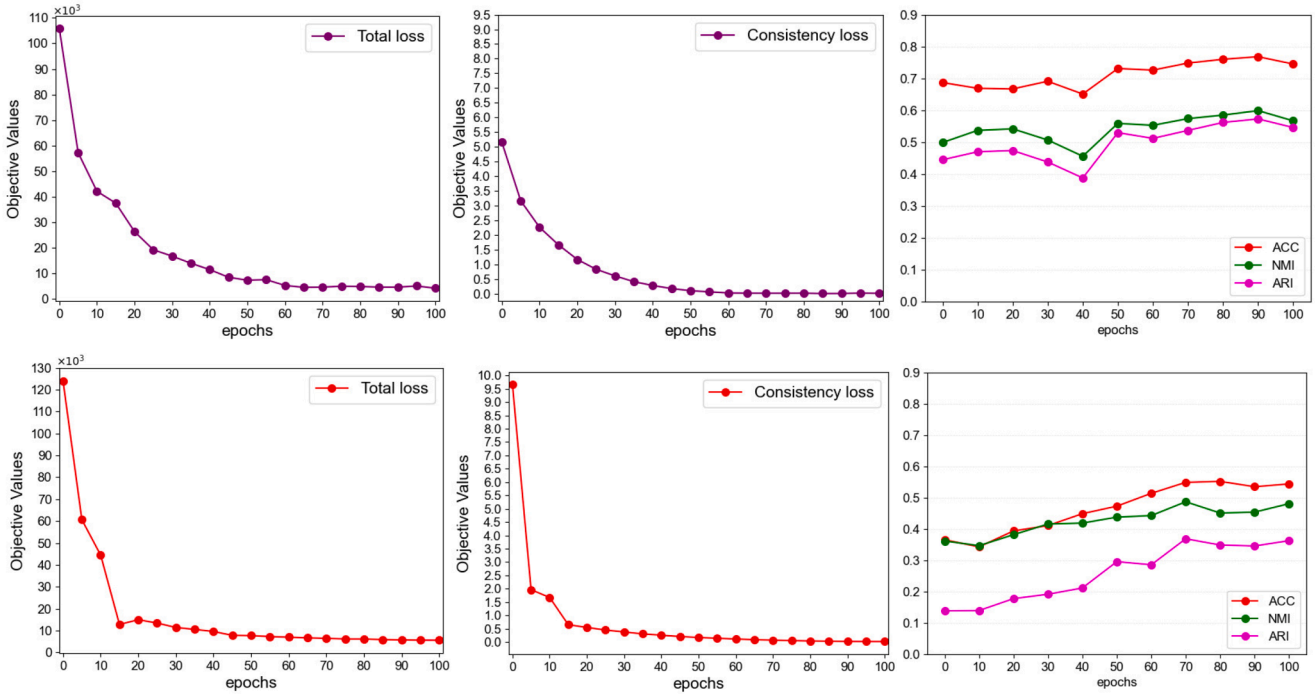

Fig. 8. The convergence curves of the total loss and consistency loss, and the clustering performance of MvCDSC on the Cora and Wiki datasets. (a-c) The results on Cora. (d-f) The results on Wiki.

4.5. Computation time

To evaluate the efficiency of the proposed method, we compare MvCDSC with four representative methods on both small (Cora) and large (HHAR) datasets. Based on the availability of source code and runtime environment, we select MAGCN, MGCCN, MFCGC and SGCMC for the comparison. The former three methods are representation-based deep multi-view clustering methods. In contrast, SGCMC and our proposed MvCDSC is a subspace-based deep multi-view clustering method. We measure the computation time until each method reaches its optimal solution. From the experimental results (Table 9), we observe that on the small dataset, the computation times for each method were

similar. However, on the large dataset, representation-based DMVC methods demonstrated advantages in terms of efficiency. MvCDSC requires additional time for implementing spectral clustering, which in turn provides higher metrics and improved stability. Nonetheless, this time investment is generally acceptable, as the suboptimal method MFCGC, requires a similar amount of time as our model.

Also, we conduct computational complexity analysis for the proposed MvCDSC and other state-of-the-art methods (see the subsection "Complexity comparison" in Supplementary analysis). As shown in Supplementary Table 3, while MvCDSC incorporates an additional shared graph autoencoder, the overall framework maintains remarkable simplicity and computational efficiency.

4.6. Visualization of clustering results

To intuitively validate the effective of McCDSC, we implement t-SNE on the learned C_{final} at three different epochs on Citeseer and ACM datasets, where different colors indicates different cluster labels. As shown in Fig. 7, as the number of epochs increases, the inter-cluster gaps between subspace representations of different clusters widen, while similar subspace representations gradually converge. The emergence of intra-cluster gaps is unavoidable due to the inherent nature of subspace representation construction. These results demonstrate that MvCDSC effectively meets our clustering needs. The results further indicate that MvCDSC effectively leverages multi-view consistency and complementarity to learn a discriminative self-expression coefficient matrix C_{final}, improving clustering results.

4.7. Convergence analysis

To validate the multi-scale consistency mechanism, we performed quantitative analyses of both the total loss and consistency loss L_{sc} across the Cora and Wiki datasets. As illustrated in Fig. 8(a,c), the total loss converges after 60 epochs (Cora) and 50 epochs (Wiki) with characteristic rapid initial decrease followed by asymptotic stabilization. Meanwhile, Lsc exhibits similar convergence dynamics to the total loss, confirming its tight coupling with the overall optimization process (Fig. 8(b,d)).Further correlational analysis (Fig. 8(c, f)) reveals that clustering metrics (ACC/NMI/ARI) show marked improvement concurrent with the convergence of Lsc and total loss despite early-training fluctuations. This empirically verifies that Lsc actively governs latent representation learning for clustering, and multi-scale consistency directly contributes to final performance.

5. Conclusion

In this article, we propose MvCDSC, a contrastive deep subspace clustering framework for multi-view graph data. MvCDSC utilizes view-specific graph auto-encoders and a shared graph auto-encoder to capture the intricacies of each view while exploring shared information across views. Different from prior DSC methods focusing on single-scale self-expression or embedding-level contrastive learning, MvCDSC introduces multi-scale consistency directly on self-expression matrices across layers, enforcing alignment between shallow and deep self-expression matrices. The Shallow layers preserve neighborhood structures while the deep layers capture global semantics. To mitigate heterogeneity gaps and semantic disparities, we introduce subspace-aware contrastive learning operating directly on self-expression coefficients, adopt false-negative exclusion strategy to mitigate sparse connectivity errors. This strategy align semantic affinities rather than raw features, which is more aligned with the clustering objective. MvCDSC unifies representation learning, subspace modeling, and contrastive alignment in a single end-to-end framework. This framework balances consistency, complementarity, and multi-scale representation learning, with potential applications in recommendation systems, bioinformatics, and social network analysis. Extensive experimental results demonstrate that MvCDSC outperforms state-of-the-art methods in node clustering tasks. Meanwhile, MvCDSC exhibits s strong generalization across diverse data types, including non-graph datasets like Caltech101 and multilayer graphs (ACM/IMDB), underscoring its versatility in handling view heterogeneity. For future work, we aim to extend the application of MvCDSC to datasets with a greater number of views and to enhance its computational efficiency.

CRediT authorship contribution statement

Yanglan Gan: Investigation, Conceptualization, Methodology, Writing – review & editing; **Zhengtian Gu:** Methodology, Software, Data curation, Formal analysis, Validation, Writing – original draft; **Guangwei Xu:** Writing – review & editing; **Guobing Zou:** Visualization, Writing – review & editing.

Data availability

Data will be made available on request.

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.

Acknowledgments

This work was sponsored in part by the National Natural Science Foundation of China (62172088, 62572114) and Shanghai Natural Science Foundation (21ZR1400400).

Supplementary material

Supplementary material associated with this article can be found in the online version at 10.1016/j.eswa.2025.130921

References

ABUD-ALLAH, H. (2024). Multi-view deep learning-based Covid-19 diagnosis with chest x-ray images: A comparative study of SVM and KNN classifiers. *EDRAAK: Peninsula Publishing Press*, *2024*, 59–77.

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

Cao, X., Zhang, C., Fu, H., Liu, S., & Zhang, H. (2015). Diversity-induced multi-view subspace clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 586–594).

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In *International conference on machine learning* (pp. 1597–1607). PMLR.

Cheng, J., Wang, Q., Tao, Z., Xie, D., & Gao, Q. (2021). Multi-view attribute graph convolution networks for clustering. In *Proceedings of the twenty-ninth international conference on international joint conferences on artificial intelligence* (pp. 2973–2979).

Cheng, L., Chen, Y., & Hua, Z. (2022). Deep contrastive multi-view subspace clustering. In *International conference on neural information processing* (pp. 692–704). Springer.

Du, S., Cai, Z., Wu, Z., Pi, Y., & Wang, S. (2024). UMCGL: Universal multi-view consensus graph learning with consistency and diversity. *IEEE Transactions on Image Processing*, *33*, 3399–3412.

Fan, S., Wang, X., Shi, C., Lu, E., Lin, K., & Wang, B. (2020). One2multi graph autoencoder for multi-view graph clustering. In *Proceedings of the web conference 2020* (pp. 3070–3076).

Fang, U., Li, M., Li, J., Gao, L., Jia, T., & Zhang, Y. (2023). A comprehensive survey on multi-view clustering. *IEEE Transactions on Knowledge and Data Engineering*, *35*(12), 12350–12368.

Gan, Y., Liu, W., Xu, G., Yan, C., & Zou, G. (2023). DMFDDI: Deep multimodal fusion for drug–drug interaction prediction. *Briefings in Bioinformatics*, *24*(6), bbad397.

He, K., Fan, H., Wu, Y., Xie, S., & Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 9729–9738).

Hu, S., Zhang, C., Zou, G., Lou, Z., & Ye, Y. (2025). Deep multiview clustering by pseudo-label guided contrastive learning and dual correlation learning. *IEEE Transactions on Neural Networks and Learning Systems*, *36*(2), 3646–3658.

Jiang, Y., Huang, C., & Huang, L. (2023). Adaptive graph contrastive learning for recommendation. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining* (pp. 4252–4261).

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kipf, T. N., & Welling, M. (2016). Variational graph auto-encoders. arXiv preprint arXiv:1611.07308.

Lele, F., Lei, Z., Jinghua, Y., Chuan, C., Chuanfu, Z., & Zibin, Z. (2022). Subspace-contrastive multi-view clustering. arXiv preprint arXiv:2210.06795.

Lin, Z., & Kang, Z. (2021). Graph filter-based multi-view attributed graph clustering. In *IJCAI* (pp. 2723–2729).

Ling, Y., Chen, J., Ren, Y., Pu, X., Xu, J., Zhu, X., & He, L. (2023). Dual label-guided graph refinement for multi-view graph clustering. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 8791–8798). (*vol. 37*).

Liu, J., Cao, F., Jing, X., & Liang, J. (2024). Deep multi-view graph clustering network with weighting mechanism and collaborative training. *Expert Systems with Applications*, *236*, 121298.

Liu, J., Wang, C., Gao, J., & Han, J. (2013). Multi-view clustering via joint nonnegative matrix factorization. In *Proceedings of the 2013 SIAM international conference on data mining* (pp. 252–260). SIAM.

Liu, L., Kang, Z., Ruan, J., & He, X. (2022). Multilayer graph contrastive clustering network. *Information Sciences*, *613*, 256–267.

Liu, W., Chen, P.-Y., Yeung, S., Suzumura, T., & Chen, L. (2017). Principled multilayer network embedding. In *2017 IEEE International conference on data mining workshops (ICDMW)* (pp. 134–141). IEEE.

Nie, F., Li, J., Li, X. et al. (2017). Self-weighted multiview clustering with multiple graphs. In *IJCAI* (pp. 2564–2570).

Pan, E., & Kang, Z. (2021). Multi-view contrastive graph clustering. *Advances in Neural Information Processing Systems*, *34*, 2148–2159.

Peng, Z., Huang, W., Luo, M., Zheng, Q., Rong, Y., Xu, T., & Huang, J. (2020). Graph representation learning via graphical mutual information maximization. In *Proceedings of the web conference 2020* (pp. 259–270).

Salehi, A., & Davulcu, H. (2019). Graph attention auto-encoders. arXiv preprint arXiv:1905.10715.

Sun, D., Li, D., Ding, Z., Zhang, X., & Tang, J. (2022). A2AE: Towards adaptive multi-view graph representation learning via all-to-all graph autoencoder architecture. *Applied Soft Computing*, *125*, 109193.

Sun, M., Wang, S., Zhang, P., Liu, X., Guo, X., Zhou, S., & Zhu, E. (2021). Projective multiple kernel subspace clustering. *IEEE Transactions on Multimedia*, *24*, 2567–2579.

Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., & Mei, Q. (2015). Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web* (pp. 1067–1077).

Velickovic, P., Fedus, W., Hamilton, W. L., Liò, P., Bengio, Y., & Hjelm, R. D. (2019). Deep graph infomax. *ICLR (Poster)*, *2*(3), 4.

Wang, C., Pan, S., Hu, R., Long, G., Jiang, J., & Zhang, C. (2019a). Attributed graph clustering: A deep attentional embedding approach. arXiv preprint arXiv:1906.06532.

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

Wang, H., Yang, Y., & Liu, B. (2019b). GMC: Graph-based multi-view clustering. *IEEE Transactions on Knowledge and Data Engineering*, *32*(6), 1116–1129.

Wang, T., Qi, Y., Qi, X., Zhang, Y., Guan, J., & Yang, G.. Contrastive representation learning for attributed graph clustering. *Available at SSRN 4424706*.

Wang, T., Wu, J., Zhang, Z., Zhou, W., Chen, G., & Liu, S. (2021a). Multi-scale graph attention subspace clustering network. *Neurocomputing*, *459*, 302–314.

Wang, Y., Chang, D., Fu, Z., & Zhao, Y. (2021b). Consistent multiple graph embedding for multi-view clustering. *IEEE Transactions on Multimedia*, *25*, 1008–1018.

Xia, R., Pan, Y., Du, L., & Yin, J. (2014). Robust multi-view spectral clustering via low-rank and sparse decomposition. In *Proceedings of the AAAI conference on artificial intelligence*. (*vol. 28*).

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*, *24*, 3182–3192.

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.

Xiao, S., Du, S., Chen, Z., Zhang, Y., & Wang, S. (2023). Dual fusion-propagation graph neural network for multi-view clustering. *IEEE Transactions on Multimedia*, *25*, 9203–9215.

Xu, C., Li, Z., Guan, Z., Zhao, W., Song, X., Wu, Y., & Li, J. (2023). Unbalanced multi-view deep learning. In *Proceedings of the 31st ACM international conference on multimedia* (pp. 3051–3059).

Xu, J., Ren, Y., Li, G., Pan, L., Zhu, C., & Xu, Z. (2021). Deep embedded multi-view clustering with collaborative training. *Information Sciences*, *573*, 279–290.

Yang, C., Liu, Z., Zhao, D., Sun, M., & Chang, E. Y. (2015). Network representation learning with rich text information. In *IJCAI* (pp. 2111–2117). (*vol. 2015*).

Yang, S., Liao, Z., Chen, R., Lai, Y., & Xu, W. (2024). Multi-view fair-augmentation contrastive graph clustering with reliable pseudo-labels. *Information Sciences*, *674*, 120739.

You, Y., Chen, T., Sui, Y., Chen, T., Wang, Z., & Shen, Y. (2020). Graph contrastive learning with augmentations. *Advances in Neural Information Processing Systems*, *33*, 5812–5823.

Zhang, Y., Liu, Y., Xu, Y., Xiong, H., Lei, C., He, W., Cui, L., & Miao, C. (2022). Enhancing sequential recommendation with graph contrastive learning. In *Proceedings of the 31st international joint conference on artificial intelligence* (pp. 2398—2405).

Zhao, M., Yang, W., & Nie, F. (2023a). Deep graph reconstruction for multi-view clustering. *Neural Networks*, *168*, 560–568.

Zhao, M., Yang, W., & Nie, F. (2023b). Deep multi-view spectral clustering via ensemble. *Pattern Recognition*, *144*, 109836.

Zhou, H., Gong, M., Wang, S., Gao, Y., & Zhao, Z. (2023). SMGCL: Semi-supervised multi-view graph contrastive learning. *Knowledge-Based Systems*, *260*, 110120.

Zhu, P., Hui, B., Zhang, C., Du, D., Wen, L., & Hu, Q. (2019). Multi-view deep subspace clustering networks. arXiv preprint arXiv:1908.01978.

Zhu, Y., Xu, Y., Yu, F., Liu, Q., Wu, S., & Wang, L. (2020). Deep graph contrastive representation learning. arXiv preprint arXiv:2006.04131.