Multi-view Comprehensive Graph Clustering

Yanying Mei, Zhenwen Ren, *Member, IEEE*, Bin Wu, Tao Yang, Yanhua Shao

***Abstract*—Graph learning can characterize the local structure relation of complex data, which has been extensively used in multi-view clustering (MVC). Currently, existing multi-view graph clustering (MVGC) methods learn the similarity of directly connected samples for clustering. However, these MVGC methods can not fully consider the indirect relation among samples and high-order relation across multi-view data. In this paper, a new multi-view comprehensive graph clustering (MCGC) method is devised, which can fully learn the similarity based on (1) first-order proximity (FOP) (*i.e.*, the direct relation of pairwise samples); (2) second-order proximity (SOP) (*i.e.*, the indirect relation of pairwise samples); and (3) third-order proximity (TOP) (*i.e.*, the three-order relation of multiple views). Since the operations of these three components are iteratively carried out, the interaction between similarity learning can be encouraged and the comprehensive graph can be generated effectively for clustering. In-depth experiments on six commonly benchmark datasets show the superiority of the MCGC method.**

***Index Terms*—Comprehensive graph learning, multi-view clustering, similarity learning, low-rank tensor representation.**

I. Introduction

Along with the diversification of data acquisition tools, data acquired from multiple sources may contain multiple heterogeneous attributes. For example, a picture can be described by color, contour, edge, etc; a news item can be reported in several characters and languages; a video consists of audio and images. All these are called *multi-view data*. As a typical processing approach for multi-view data, multi-view clustering (MVC) has become a popular topic. MVC attempts to divide multi-view data into various clusters to ensure that samples are highly correlated inside the same cluster [1, 2] and has been widely used in consumer electronics [3–5], computer science [6–8], statistics [9, 10], and other fields.

In the real scene, data samples have amounts of relation graphs as these samples in each view can construct a relation graph. To effectively learn the data points relation for clustering, numerous graph-based MVC methods (MVGC) have been proposed. The MVGC methods generally learn a fusion affinity graph from multiple input graphs to explore the essential structure of multi-view data for clustering [11, 12]. However, these methods generally learn the affinity graphs by the direct similarity learning between pairwise data points, and the learned affinity graph from each view is isolated or fixed during fusion. Some improved methods [13, 14] are proposed. The work [13] is developed to learn each input affinity graph and fusion affinity graph interactively for clustering. The work [14] is proposed to learn a robust latent embedding representation of multiple views and perform spectral clustering (SPC) in learned latent embedding space. Despite these MVGC approaches have obtained better clustering performance, they usually learn the affinity graph based on first-order proximity (FOP) and do not consider the indirect relation of pairwise data points, ignoring that the relation among data points should be propagative.

In the area of information networks, some scholars have proposed a network based on second-order proximity (SOP) to enhance network performance [15]. From the viewpoint of the graph, it represents the affinity graph based on SOP which can help excavate the indirect relation among data points, theoretically, it can promote graph clustering. Moreover, most current MVGC methods [14, 16] focus on matrix-oriented operations, which are modeled by matrix theory. They fail to fully excavate view consistency across multiple views. Because tensor learning can excavate the high-order relation across multiple representations, some tensor-based approaches [17–20] are suggested, in which the similarity based on the three-order proximity (TOP) is learned. Nevertheless, these approaches can not comprehensively consider the information of the graph, resulting in the learned affinity graph not being optimal.

To learn a comprehensive affinity graph, in our paper, a new MVGC method is suggested, namely multi-view comprehensive graph clustering (MCGC). Figure 1 presents an overview of MCGC. Specifically, the FOP and SOP similarity are learned to explore the direct and indirect relation among different samples, and the TOP similarity is learned to explore the three-order correlation across multi-view data. Subsequently, a comprehensive graph is obtained through the collaborative learning based on FOP, SOP, and TOP, and SPC is used on the learned comprehensive graph to achieve the final clustering result. The main contributions are highlighted as follows:

- A new comprehensive graph learning method is proposed. Unlike the similarity learning of directly connected samples in existing methods, comprehensive graph learning can fully learn the similarity based on FOP, SOP and TOP to obtain an optimal consistency affinity graph.
- The similarity learning based on FOP, SOP and TOP is devised to excavate the direct and indirect relation among samples, and the high-order relation across multi-view data. These three parts can be mutually promoted to achieve the best clustering result.
- We validate the effectiveness of MCGC experimentally on six benchmark databases in various applications. The results notably show its superiority over eight other related state-of-the-art methods for clustering.

This research was supported by the Sichuan Science and Technology Program (Grant nos. 2019ZDZX0043 and 2020ZDZX0014).

Zhenwen Ren is with the Department of National Defence Science and Technology, Southwest University of Science and Technology, Mianyang, China, 621010 (e-mail: rzw@njust.edu.cn).

Zhenwen Ren is the corresponding author.

The rest of our paper is organized as follows. Some closely related clustering methods are introduced in Section II. Section III give some notation definitions and preliminaries about tensor. Section IV proposes our MCGC model, and then gives the optimization and complexity analysis of our algorithm. The experimental analysis and results are discussed in Section V. Finally, some conclusions are derived in Section VI.

II. Related work

As the fundamental inspiration for the suggested method, two types of clustering methods are briefly reviewed: (1) graph-based clustering and (2) tensor-based clustering.

As aforementioned, graph-based clustering [11, 21] excavates the local structure relation by constructing the relation graph among data points for clustering. Given N data points set $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N] \in \mathbb{R}^{D\times N}$, the graph-based clustering model is defined as

$$\min_{\mathbf{Z}} \sum_{\substack{i=1,\\ j=1}}^{N} f(\mathbf{x}_i, \mathbf{x}_j) z_{ij} + \alpha \mathcal{R}(\mathbf{Z}) \quad \text{s.t. } \mathbf{Z}^\top = \mathbf{Z}, \mathbf{0} \preceq \mathbf{Z} \preceq \mathbf{1}, \tag{1}$$

where the function f is usually represented by Euclidean distance (*i.e.*, $\|\mathbf{x}_i - \mathbf{x}_j\|_2^2$) [11] between two data points (*i.e.*, $\mathbf{x}_i$ and $\mathbf{x}_j$), and a smaller distance represents a larger similarity z_{ij}; $\mathcal{R}(\mathbf{Z})$ is the regularization about $\mathbf{Z}$ (such as nuclear norm [22], sparse norm [12], Schatten-p norm [23], Frobenius norm [24], and block diagonal constraint [25]), and α is the regularization parameter; z_{ij} is the element in $\mathbf{Z}$, and the simplex constraint (*i.e.*, $\mathbf{Z}^\top = \mathbf{Z}$, $\mathbf{0} \preceq \mathbf{Z} \preceq \mathbf{1}$) on $\mathbf{Z}$ is to guarantee the similarity of $\mathbf{Z}$.

Recently, the tensor-based clustering model is adopted to excavate the high-order correlation across multiple views. Given the multiple affinity graphs $\mathbf{Z}^{(v)}$ from the multiple views, the tensor $\boldsymbol{\mathcal{Z}}$ is obtained by merging all the affinity graphs and the low-rank constraint is used to explore view consistency across different views. For the tensor singular value decomposition (t-SVD) based on tensor nuclear norm (TNN) $\|\boldsymbol{\mathcal{Z}}\|_{\circledast}$ minimization can approximate the tensor low-rank constraint, therefore, the tensor-based clustering model is written as

$$\min_{\mathbf{Z}^{(v)}} \sum_{v=1}^{V} \sum_{\substack{i=1,\\ j=1}}^{N} g(\mathbf{x}_i^{(v)}, \mathbf{x}_j^{(v)}, z_{ij}^{(v)}) + \beta \|\boldsymbol{\mathcal{Z}}\|_{\circledast} \tag{2}$$
$$\text{s.t. } \left(\mathbf{Z}^{(v)}\right)^\top = \mathbf{Z}^{(v)}, \mathbf{0} \preceq \mathbf{Z}^{(v)} \preceq \mathbf{1},$$

where $z_{ij}^{(v)}$ in affinity graph $\mathbf{Z}^{(v)}$ represents the the affinity among data points (*e.g.*, $\mathbf{x}_i$ and $\mathbf{x}_j$) in the v-th view, the function g represents the relation among data points and $z_{ij}^{(v)}$, β is the parameter, $\boldsymbol{\mathcal{Z}}$ represents the tensor that is constructed by stacking all $\mathbf{Z}^{(v)}$, and $\|\cdot\|_{\circledast}$ represents the nuclear norm constraint. The tensor-based MVGC methods [26–29] mainly concentrate on the low-rank tensor construction. For example, the work [30] directly stacks multiple affinity matrices into a 3-order tensor and the TNN constraint is used to explore the high-order correlation across multiple representations. The work [31] firstly use TNN constraint on the rotated tensor to better excavate the high-order correlation across multiple representations. Subsequently, some improved methods based on the rotated tensor are proposed to further improve clustering performance. One representative method is the essential tensor learning for SPC (ETLMSC) [32], which concentrats on learning the transition probability relations among multiple input graphs. In work [33], multiple low-rank tensors constrained representation matrices and consensus indicator matrices learn from each other in a unified framework for MVC. These issues drive us to seek an efficient and effective MVGC method.

III. Notations and preliminaries

A. Notations

Here, we give a summary of the notations and tensor preliminaries used in this paper. The bold calligraphy letters (*e.g.*, $\boldsymbol{\mathcal{Q}} \in \mathbb{R}^{N_1\times N_2\times N_3}$), capital letters (*e.g.*, $\mathbf{Q}$), and small letters (*e.g.*, $\mathbf{q}$) denote tensor, matrix, and vector, respectively. q_{ij} and the Greek letters (*e.g.*, α, β, λ) denote the (i,j)-th entry of matrix $\mathbf{Q}$ and the scalars, respectively. For the tensor $\boldsymbol{\mathcal{Q}} \in \mathbb{R}^{N_1\times N_2\times N_3}$, $\boldsymbol{\mathcal{Q}}^{(k)}$ or $\mathbf{Q}^{(k)}$ represents the k-th frontal slice with $k = 1, \ldots, N_3$. Specifically, $\boldsymbol{\mathcal{Q}}_f = \text{fft}(\boldsymbol{\mathcal{Q}}, [\,], 3)$ and $\boldsymbol{\mathcal{Q}} = \text{ifft}(\boldsymbol{\mathcal{Q}}_f, [\,], 3)$ denote the fast Fourier transformation (FFT) and inverse FFT along the third direction of tensor $\boldsymbol{\mathcal{Q}}$, respectively. $\text{bvec}(\boldsymbol{\mathcal{Q}}) = \left[\mathbf{Q}^{(1)}; \mathbf{Q}^{(2)}; \cdots; \mathbf{Q}^{(N_3)}\right] \in \mathbb{R}^{N_1N_3\times N_2}$ and $\text{fold}(\text{bvec}(\boldsymbol{\mathcal{Q}})) = \boldsymbol{\mathcal{Q}}$ denote the block vectorization and the corresponding inverse operation, respectively. $\text{bdiag}(\boldsymbol{\mathcal{Q}}) \in \mathbb{R}^{N_1N_3\times N_2N_3}$ and $\text{bcirc}(\boldsymbol{\mathcal{Q}}) \in \mathbb{R}^{N_1N_3\times N_2N_3}$ represent the block diagonal matrix and the corresponding block circular matrix, respectively. The f-diagonal tensor satisfies that all its frontal slices are diagonal. The identity tensor $\boldsymbol{\mathcal{I}} \in \mathbb{R}^{N_1\times N_1\times N_3}$ satisfies that its first frontal slice is a $N_1 \times N_1$ identity matrix, and all subsequent frontal slices are zero matrices.

B. Preliminaries

To better comprehend the tensor operations, here, we present some tensor preliminaries used in the paper.

Definition 1 (3-order Tensor). The 3-order tensor $\boldsymbol{\mathcal{Q}} \in \mathbb{R}^{N_1\times N_1\times V}$ is obtained by merging all the V matrices $\mathbf{Q}^{(v)} \in \mathbb{R}^{N_1\times N_1}$, which is given by

$$\boldsymbol{\mathcal{Q}} = \text{bvfold}([\mathbf{Q}^{(1)}; \cdots; \mathbf{Q}^{(V)}]). \tag{3}$$

Definition 2 (Tensor Rotation). The tensor rotation for $\boldsymbol{\mathcal{Q}} \in \mathbb{R}^{N_1\times N_1\times N_3}$ is given by

$$\boldsymbol{\mathcal{Q}}^{*} = \text{rotate}(\boldsymbol{\mathcal{Q}}), \tag{4}$$

where $\boldsymbol{\mathcal{Q}}^{*} \in \mathbb{R}^{N_1\times N_3\times N_1}$. Keep in mind that the shift function rotate is used here.

Definition 3 (T-Product). The tensor t-product is given as

$$\boldsymbol{\mathcal{Q}} * \boldsymbol{\mathcal{P}} = \text{fold}(\text{bcirc}(\boldsymbol{\mathcal{Q}})\,\text{bvec}(\boldsymbol{\mathcal{P}})). \tag{5}$$

Definition 4 (Orthogonal Tensor). The tensor $\boldsymbol{\mathcal{Q}}$ is orthogonal if it satisfies

$$\boldsymbol{\mathcal{Q}}^{\mathrm{T}} * \boldsymbol{\mathcal{Q}} = \boldsymbol{\mathcal{Q}} * \boldsymbol{\mathcal{Q}}^{\mathrm{T}} = \boldsymbol{\mathcal{I}}. \tag{6}$$

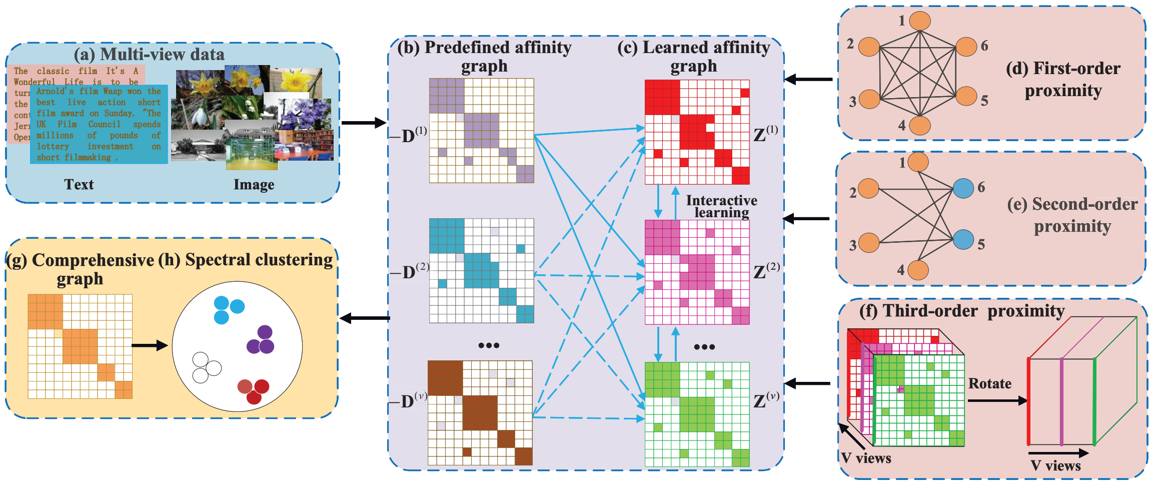


Fig. 1. The overview of our proposed MCGC. (a) For multi-view data, (b) V predefined affinity graphs are given; (c) V learned affinity graphs corresponding to multiple views are built; (d) the first-order proximity similarity is learned; (e) the second-order proximity similarity is learned; (f) the third-order proximity similarity is learned; (g) the comprehensive graph $\bar{\mathbf{Z}}$ is achieved by averaging all learned affinity graphs $\mathbf{Z}^{(v)}$; and (h) the SPC is used to achieve the clustering result.

Definition 5 (t-SVD). The t-SVD can be expressed as

$$\mathcal{Q} = \mathcal{U} * \mathcal{S} * \mathcal{V}^{\top}, \tag{7}$$

where $\mathcal{S} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ is f-diagonal, and $\mathcal{U} \in \mathbb{R}^{N_1 \times N_1 \times N_3}$, $\mathcal{V} \in \mathbb{R}^{N_2 \times N_2 \times N_3}$ are orthogonal.

Definition 6 (TNN). The TNN $\|\mathcal{Q}\|_{\circledast}$ is defined as

$$\|\mathcal{Q}\|_{\circledast} = \sum_{k=1}^{N_3} \left\|\mathcal{Q}_f^{(k)}\right\|_{*} = \sum_{i=1}^{\min(N_1,N_2)} \sum_{k=1}^{N_3} \left|\mathcal{S}_f^{(k)}(i,i)\right|, \tag{8}$$

where $\mathcal{S}_f^{(k)}$ is computed by the t-SVD of $\mathcal{Q}_f^{(k)} = \mathcal{U}_f^{(k)} \mathcal{S}_f^{(k)} \mathcal{V}_f^{(k)\top}$.

Definition 7 (Tensor Transpose). The tensor $\mathcal{Q} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ transpose is achieved by transposing all its frontal slices, which is represented by $\mathcal{Q}^{\top} \in \mathbb{R}^{N_2 \times N_1 \times N_3}$.

IV. Proposed Method

Now we present our MCGC method which includes the similarity learning based on FOP, SOP, and TOP. The main theories and formulas of each part are presented in this sections. The optimization and complexity analysis are also presented later.

A. Similarity Learning Based on First-Order Proximity

The FOP describes the direct affinity among pairwise data points, which is the first and foremost measures of similarity. Given $\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(V)}$ to represent V views in a multi-view dataset, in which $\mathbf{X}^{(v)} = \left\{\mathbf{x}_1^{(v)}, \ldots, \mathbf{x}_N^{(v)}\right\} \in \mathbb{R}^{d_v \times N}$ represents the v-th view with N data points and dimensionality of d_v. For the constructed affinity graph, a smaller distance between pairwise data points denotes a large similarity, and a larger distance between pairwise data points denotes a small (or zero) similarity. Therefore, the similarity learning model based on FOP is given by

$$\begin{aligned} &\min_{\mathbf{Z}^{(v)}} \sum_{i,j=1}^{N} \left\|\mathbf{x}_i^{(v)} - \mathbf{x}_j^{(v)}\right\|_2^2 z_{ij}^{(v)} \\ &\text{s.t. } \mathbf{Z}^{(v)\top} = \mathbf{Z}^{(v)}, \mathbf{0} \preceq \mathbf{Z}^{(v)} \preceq \mathbf{1}, \end{aligned} \tag{9}$$

where $z_{ij}^{(v)}$ in affinity graph $\mathbf{Z}^{(v)} \in \mathbb{R}^{N \times N}$ represents the affinity between $\mathbf{x}_i^{(v)}$ and $\mathbf{x}_j^{(v)}$.

Inspired by [21], we have

$$\begin{aligned} &\min_{\mathbf{Z}^{(v)}} \sum_{i,j=1}^{N} \left\|\mathbf{x}_i^{(v)} - \mathbf{x}_j^{(v)}\right\|_2^2 z_{ij}^{(v)} = \min_{\mathbf{Z}^{(v)}} \operatorname{Tr}(\mathbf{D}^{(v)\top}\mathbf{Z}^{(v)}) \\ &\text{s.t. } \mathbf{Z}^{(v)\top} = \mathbf{Z}^{(v)}, \mathbf{0} \preceq \mathbf{Z}^{(v)} \preceq \mathbf{1}, \end{aligned} \tag{10}$$

where $\mathbf{D}^{(v)}$ is called the v-th complete graph with $d_{ij}^{(v)} = \left\|\mathbf{x}_i^{(v)} - \mathbf{x}_j^{(v)}\right\|_2^2$. A smaller value of $d_{ij}^{(v)}$ represents a larger value of similarity $z_{ij}^{(v)}$.

To minimize the disagreement among different views, given the v-th affinity graph $\mathbf{Z}^{(v)}$ and the w-th affinity graph $\mathbf{Z}^{(w)}$, we can constrain the error term $\left\|\mathbf{Z}^{(v)} - \mathbf{Z}^{(w)}\right\|_F^2$. In comparison with direct similarity discrepancy $\left\|\mathbf{Z} - \mathbf{Z}^{(v)}\right\|_F^2$, the penalization on the interactive learning considers the consistency between pairwise views. Consequently, the learned affinity graph becomes noise-resistant. Therefore, Eq. (10) can be transformed into

$$\min_{\mathbf{Z}^{(v)}} \sum_{v=1}^{2V} \sum_{\substack{w=1,\\ w\neq v}}^{2V} \left\|\mathbf{Z}^{(v)} - \mathbf{Z}^{(w)}\right\|_F^2. \tag{11}$$

The affinity graph $\mathbf{Z}^{(v)}$ is given as follows: if $1 \leq v \leq V$, the learned affinity graph is defined as $\mathbf{Z}^{(v)}$; if $(1+V) \leq v \leq 2V$, the predefined affinity graph is given by $-\mathbf{D}^{(v-V)}$.

B. Similarity Learning Based on Second-Order Proximity

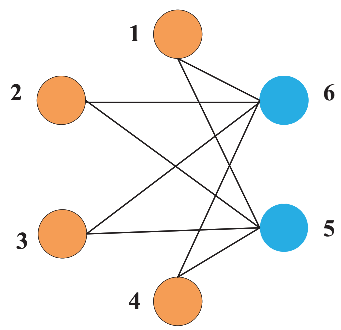


Fig. 2. The example of the SOP. Data points 5 and 6 should also be similar as they have the common neighbors set $\{1, 2, \cdots, 4\}$.

The SOP describes the proximity of the neighborhood structures of pairwise data points, which aims to excavate the indirect relation of data points. For the v-th view, given the neighbors of data point $\mathbf{x}_i^{(v)}$, denoted as $\mathbf{x}_j^{(v)} \in \mathbf{X}^{(v)}$, $j = 1, \ldots, N$, $z_{ij}^{(v)}$ denotes the affinity between $\mathbf{x}_i^{(v)}$ and $\mathbf{x}_j^{(v)}$. Then, the SOP similarity learning between data points is written as

$$\begin{aligned} &\min_{\mathbf{Z}^{(v)}} \sum_{i=1}^{N} \left\| \mathbf{x}_i^{(v)} - \sum_{j=1}^{N} z_{ij}^{(v)} \mathbf{x}_j^{(v)} \right\|_2^2 \\ &\text{s.t. } \mathbf{Z}^{(v)\top} = \mathbf{Z}^{(v)}, \mathbf{0} \preceq \mathbf{Z}^{(v)} \preceq \mathbf{1}. \end{aligned} \tag{12}$$

With Eq. (12), we can observe that the data point $\mathbf{x}_i^{(v)}$ is close to its neighbors. From Fig. 2, data points 5 and 6 have the common neighbors set $\{1, 2, \cdots, 4\}$, Eq. (12) ensures that data point 5 and 6 are close to the same data points set $\{1, 2, \cdots, 4\}$. As a result, even though data points 5 and 6 are not directly connected, they will be close. This maintains the relation of two disconnected data points.

According to [34], Eq. (12) can be rewritten as

$$\begin{aligned} &\min_{\mathbf{Z}^{(v)}} \mathrm{Tr}\left(\mathbf{X}^{(v)}(\mathbf{I} - \mathbf{Z}^{(v)})(\mathbf{I} - \mathbf{Z}^{(v)})^\top \mathbf{X}^{(v)\top}\right) \\ &\text{s.t. } \mathbf{Z}^{(v)\top} = \mathbf{Z}^{(v)}, \mathbf{0} \preceq \mathbf{Z}^{(v)} \preceq \mathbf{1}, \end{aligned} \tag{13}$$

where $\mathbf{X}^{(v)} = [\mathbf{x}_1^{(v)}, \mathbf{x}_2^{(v)}, \cdots, \mathbf{x}_N^{(v)}]$ represents the v-th view with N data points and the matrix's trace is denoted by $\mathrm{Tr}(\cdot)$.

C. Similarity Learning Based on Third-Order Proximity

To well explore the view consistency across multi-view representations [32]. A third-order tensor $\boldsymbol{\mathcal{Z}}$ based model is proposed. The convex TNN $\|\cdot\|_{\circledast}$ which is described in Definition 6 is adopted to approximate the low-rank tensor representation. Thus, the similarity learning model based on TOP (*i.e.*, tensor) is written as

$$\begin{aligned} &\min_{\mathbf{Z}^{(v)}} \sum_{v=1}^{V} \mathrm{Tr}(\mathbf{X}^{(v)}(\mathbf{I} - \mathbf{Z}^{(v)})(\mathbf{I} - \mathbf{Z}^{(v)})^\top \mathbf{X}^{(v)\top}) \\ &\quad + \sum_{v=1}^{2V} \sum_{\substack{w=1, \\ w \neq v}}^{2V} \alpha \left\| \mathbf{Z}^{(v)} - \mathbf{Z}^{(w)} \right\|_F^2 + \beta \|\boldsymbol{\mathcal{Z}}\|_{\circledast} \\ &\quad \text{s.t.} \quad \mathbf{Z}^{(v)\top} = \mathbf{Z}^{(v)}, \mathbf{0} \preceq \mathbf{Z}^{(v)} \preceq \mathbf{1}, \\ &\quad \boldsymbol{\mathcal{Z}} = \Phi(\mathbf{Z}^{(1)}, \cdots, \mathbf{Z}^{(V)}), \end{aligned} \tag{14}$$

where tensor $\boldsymbol{\mathcal{Z}} \in \mathbb{R}^{N \times V \times N}$ represents the rotated tensor, $\Phi(\cdot)$ represents merging all input graphs $\mathbf{Z}^{(v)}$ into a tensor and rotating the tensor, $z_{ij}^{(v)}$ represents the affinity between $\mathbf{x}_i^{(v)}$ and $\mathbf{x}_j^{(v)}$ in the v-th view, and $\alpha > 0$, $\beta > 0$ are the trade-off parameters. The rotation of the tensor is beneficial for excavating consistent information across multiple views.

Strict block diagonal structure of the affinity graph is good for clustering [35]. Hence, we wish to obtain a strict block diagonal affinity graph in our model. From [35], if the matrix $\mathbf{U}$ satisfies $\mathbf{U} := \{\mathbf{U} \mid \mathbf{U} \in \mathbb{R}^{N \times c}, \mathbf{U}^\top \mathbf{U} = \mathbf{I}\}$, $\mathbf{U}\mathbf{U}^\top$ satisfies strict block diagonal. So we can constrain the affinity graph $\mathbf{Z}$ to remain strict block diagonal according to Theorem 1.

Theorem 1 [35] *$\mathbb{Z}_1 := \left\{\mathbf{Z} = \mathbf{U}\mathbf{U}^\top \mid \mathbf{U}^\top \mathbf{U} = \mathbf{I}\right\}$, $\mathbf{U} \in \mathbb{R}^{N \times c}$, and $\mathbb{Z}_2 := \left\{\mathbf{Z} \mid \mathbf{Z} = \mathbf{Z}^\top, \mathrm{Tr}(\mathbf{Z}) = c, \mathbf{0} \leq \mathbf{Z} \leq \mathbf{1}\right\}$, $\mathbb{Z}_2$ is the convex hull of $\mathbb{Z}_1$, and $\mathbb{Z}_1$ is exactly the extreme points set of $\mathbb{Z}_2$.*

Therefore, the ideal affinity graph $\mathbf{Z}$ is obtained by

$$\begin{aligned} &\min_{\mathbf{Z}^{(v)}} \sum_{v=1}^{V} \mathrm{Tr}(\mathbf{X}^{(v)}(\mathbf{I} - \mathbf{Z}^{(v)})(\mathbf{I} - \mathbf{Z}^{(v)})^\top \mathbf{X}^{(v)\top}) \\ &\quad + \sum_{v=1}^{2V} \sum_{\substack{w=1, \\ w \neq v}}^{2V} \alpha \left\| \mathbf{Z}^{(v)} - \mathbf{Z}^{(w)} \right\|_F^2 + \beta \|\boldsymbol{\mathcal{Z}}\|_{\circledast} \\ &\quad \text{s.t. } \mathbf{Z}^{(v)\top} = \mathbf{Z}^{(v)}, \mathbf{0} \preceq \mathbf{Z}^{(v)} \preceq \mathbf{1}, \\ &\quad \boldsymbol{\mathcal{Z}} = \Phi(\mathbf{Z}^{(1)}, \cdots, \mathbf{Z}^{(V)}), \mathrm{Tr}(\mathbf{Z}^{(v)}) = c, \end{aligned} \tag{15}$$

where c denotes the number of clusters.

In summary, a comprehensive graph $\mathbf{Z}$ can be learned in Eq. (15), which comprehensively learn the similarity based on FOP, SOP and TOP among different data points. Thus, the learned comprehensive graph can well represent the clustering-friendly structure which is beneficial for improving MVC performance.

D. Optimization Procedure

The variables in Eq. (15) are all coupled together which is difficult to solve each variable. Thus, the alternating direction method of multipliers (ADMM) [35] is employed to solve this problem. We introduce the auxiliary variable $\mathcal{B}$ to transform the optimization problem as

$$\begin{aligned}
&\mathcal{L}\left(\left\{\mathbf{Z}^{(v)}\right\}_{v=1}^{V}, \mathcal{B}\right) \\
=&\sum_{v=1}^{V} \operatorname{Tr}(\mathbf{X}^{(v)}(\mathbf{I}-\mathbf{Z}^{(v)})(\mathbf{I}-\mathbf{Z}^{(v)})^{\top}\mathbf{X}^{(v)\top}) \\
&+\sum_{v=1}^{2V}\sum_{\substack{w=1,\\ w\neq v}}^{2V} \alpha \left\|\mathbf{Z}^{(v)}-\mathbf{Z}^{(w)}\right\|_F^2 + \beta\|\mathcal{B}\|_{\circledast} \\
&+\frac{\mu}{2}\left\|\mathcal{Z}-\mathcal{B}+\frac{\mathcal{Y}}{\mu}\right\|_F^2 \\
&\text{s.t. } \mathbf{Z}^{(v)\top}=\mathbf{Z}^{(v)}, \mathbf{0}\preceq\mathbf{Z}^{(v)}\preceq\mathbf{1}, \operatorname{Tr}(\mathbf{Z}^{(v)})=c, \\
&\mathcal{Z}=\Phi(\mathbf{Z}^{(1)},\cdots,\mathbf{Z}^{(V)}),
\end{aligned} \tag{16}$$

where μ represents a positive penalty parameter and $\mathcal{Y}$ denotes the Lagrangian multiplier. Next, each variable in Eq. (16) is updated alternately.

(1) **Update** $\mathbf{Z}^{(v)}$: Other variables are fixed, the sub-problem for updating $\{\mathbf{Z}^{(v)}\}_{v=1}^{V}$ is written as

$$\begin{aligned}
\min_{\mathbf{Z}^{(v)}} &\sum_{v=1}^{V} \operatorname{Tr}(\mathbf{X}^{(v)}(\mathbf{I}-\mathbf{Z}^{(v)})(\mathbf{I}-\mathbf{Z}^{(v)})^{\top}\mathbf{X}^{(v)\top}) \\
&+\sum_{v=1}^{2V}\sum_{\substack{w=1,\\ w\neq v}}^{2V} \alpha \left\|\mathbf{Z}^{(v)}-\mathbf{Z}^{(w)}\right\|_F^2 \\
&+\frac{\mu}{2}\left\|\mathbf{Z}^{(v)}-\mathbf{B}^{(v)}+\frac{\mathbf{Y}^{(v)}}{\mu}\right\|_F^2 \\
&\text{s.t. } \mathbf{Z}^{(v)\top}=\mathbf{Z}^{(v)}, \mathbf{0}\preceq\mathbf{Z}^{(v)}\preceq\mathbf{1}, \operatorname{Tr}(\mathbf{Z}^{(v)})=c, \\
&\mathcal{Z}=\Phi(\mathbf{Z}^{(1)},\cdots,\mathbf{Z}^{(V)}),
\end{aligned} \tag{17}$$

where $\mathbf{Y}^{(v)}$ and $\mathbf{B}^{(v)}$ denote the v-th slices in tensor $\mathcal{B}$ and $\mathcal{Y}$, respectively. Accordingly, Eq. (17) can be solved by

$$\begin{aligned}
\min_{\mathbf{Z}^{(v)}} &\frac{1}{2}\left\|\mathbf{Z}^{(v)}-\mathbf{P}^{(v)}\right\|_F^2 \\
&\text{s.t. } \mathbf{Z}^{(v)\top}=\mathbf{Z}^{(v)}, \mathbf{0}\preceq\mathbf{Z}^{(v)}\preceq\mathbf{1}, \operatorname{Tr}(\mathbf{Z}^{(v)})=c,
\end{aligned} \tag{18}$$

where

$$\begin{aligned}
\mathbf{P}^{(v)}=&(2\mathbf{X}^{(v)\top}\mathbf{X}^{(v)}+\mu\mathbf{I})^{-1}(\mu\mathbf{B}^{(v)}-\mathbf{Y}^{(v)} \\
&+2\mathbf{X}^{(v)\top}\mathbf{X}^{(v)}+\alpha\sum_{\substack{w=1,\\ w\neq v}}^{2V}\mathbf{Z}^{(w)}).
\end{aligned} \tag{19}$$

Eq. (18) has an optimal solution according to Theorem 2.

Theorem 2 [35] *For* $\mathbf{Z}\in\mathbb{R}^{N\times N}$, *which is a symmetric affinity matrix, the SVD for* $\mathbf{Z}$ *is given as* $\mathbf{P}=\mathbf{R}\operatorname{Diag}(\boldsymbol{\zeta})\mathbf{R}^{\top}$.

$$\min_{\mathbf{Z}} \frac{1}{2}\|\mathbf{Z}-\mathbf{P}\|_F^2 \quad s.t.\ \mathbf{Z}^{\top}=\mathbf{Z}, \mathbf{0}\preceq\mathbf{Z}^{(v)}\preceq\mathbf{1}, \operatorname{Tr}(\mathbf{Z})=c \tag{20}$$

can be solved by $\mathbf{Z}^{*}=\mathbf{R}\operatorname{Diag}(\boldsymbol{\varrho}^{*})\mathbf{R}^{\top}$, *where* $\boldsymbol{\varrho}^{*}$ *can be obtained according to*

$$\min_{\boldsymbol{\varrho}} \frac{1}{2}\|\boldsymbol{\varrho}-\boldsymbol{\zeta}\|_2^2, \quad s.t.\ 0\leq\boldsymbol{\varrho}\leq 1, \boldsymbol{\varrho}^{\top}1=c. \tag{21}$$

In the end, Eq. (21) can be solved according to [36].

(2) **Update** $\mathcal{B}$: Other variables are fixed, the sub-problem for updating $\mathcal{B}$ is written as

$$\min_{\mathcal{B}} \beta\|\mathcal{B}\|_{\circledast}+\frac{\mu}{2}\left\|\mathcal{B}-\left(\mathcal{Z}+\frac{\mathcal{Y}}{\mu}\right)\right\|_F^2. \tag{22}$$

Given $\mathcal{C}=\mathcal{Z}+\frac{\mathcal{Y}}{\mu}$, Eq. (22) can be effectively solved according to Theorem 3.

Theorem 3 [31] *Suppose* $\mathcal{B}\in\mathbb{R}^{N_1\times N_2\times N_3}$, $\mathcal{C}\in\mathbb{R}^{N_1\times N_2\times N_3}$ *with t-SVD* $\mathcal{C}=\mathcal{U}*\mathcal{M}*\mathcal{V}^{\top}$, *and scalar* $\tau>0$,

$$\min_{\mathcal{B}} \tau\|\mathcal{B}\|_{\circledast}+\frac{1}{2}\|\mathcal{B}-\mathcal{C}\|_F^2 \tag{23}$$

can be derived via the tensor tubal-shrinkage operator

$$\mathcal{B}=\mathcal{U}*\mathcal{F}_{N_3\tau}(\mathcal{M})*\mathcal{V}^{\top}, \tag{24}$$

where $\mathcal{F}_{N_3\tau}(\mathcal{M})=\mathcal{M}*\mathcal{Q}$. *The tensor* $\mathcal{Q}\in\mathbb{R}^{N_1\times N_2\times N_3}$ *satisfies* f*-diagonal with diagonal element* $\mathcal{Q}_f(i,i,j)=\left(1-\frac{N_3\tau}{\mathcal{M}(i,i,j)}\right)_+$.

(3) **Update** $\mathcal{Y}$ and μ: The Lagrange multiplier $\mathcal{Y}$ and the penalty parameter μ are updated by

$$\begin{aligned}
&\mathcal{Y}=\mathcal{Y}+\mu(\mathcal{B}-\mathcal{Z}), \\
&\mu=\min\left(\nu\mu,\mu_{\max}\right).
\end{aligned} \tag{25}$$

The convergence condition in each iteration is set as

$$\text{error } =\|\mathcal{B}-\mathcal{Z}\|_{\infty}<\epsilon, \tag{26}$$

where $\epsilon=10^{-7}$ is the convergence threshold.

Therefore, along with the initialization of all variables, our algorithm begins to update all variables alternatively until the convergence condition are satisfied. The optimization algorithm of MCGC is presented in Algorithm 1.

Algorithm 1 MCGC optimized by ADMM

Input: Multi-view data $\{\mathbf{X}^{(v)}\}_{v=1}^{V}$, parameters α and β.
1: Initialize: $\mathcal{B}=\mathcal{Z}=\mathbf{0}$, $\mu=10^{-4}$, and $\nu=1.8$.
2: **while** not converged **do**
3: Update each affinity graph $\mathbf{Z}^{(v)}$ via (17);
4: Update $\mathbf{B}^{(v)}$ via (22);
5: Update ADMM involved variables via (25);
6: **end while**
7: Perform SPC on comprehensive graph $\overline{\mathbf{Z}}$.
Output: The clustering results.

With the learned affinity graphs $\{\mathbf{Z}^{(v)}\}_{v=1}^{V}$, the comprehensive graph $\overline{\mathbf{Z}}$ with clustering-friendly structure is given as

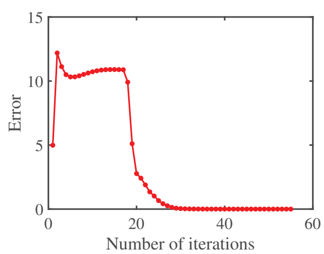
$$\overline{\mathbf{Z}}=\sum_{v=1}^{V}(|\mathbf{Z}^{(v)}|+|\mathbf{Z}^{(v)\top}|)/V. \tag{27}$$

Finally, the clustering result is obtained by performing SPC on the learned comprehensive graph.

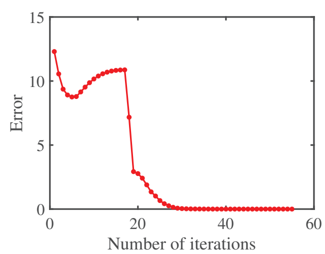
E. Complexity Analysis

For MCGC listed in Algorithm 1, the computation complexity in solving Eq. (16) is as follows: The computation complexity of updating $\mathbf{Z}^{(v)}$ is $\mathcal{O}\left(VN^2\right)$. For tensor $\boldsymbol{\mathcal{B}}$, the updating includes the tensor FFT, inverse FFT, and t-SVD. Therefore, step 4 has the total computation complexity of $\mathcal{O}\left(VN^2\log(N)+V^2N^2\right)$. The computation complexity of step 5 is $\mathcal{O}(V)$. It is theoretically that the computation complexity of MCGC is $\mathcal{O}(t(VN^2+VN^2\log(N)+V^2N^2+V))$, where t represents the iterations. Considering the computation complexity of SPC is $\mathcal{O}(N^3)$ and $V \ll N$, $t \ll N$, the total computation complexity of MCGC is $\mathcal{O}(N^3)$.

F. Convergence Analysis



(a) COIL-20



(b) Scene-15

Fig. 3. Convergence curves of MCGC on the COIL-20 and Scene-15 databases.

The convergence of MCGC listed in Algorithm 1 is analyzed in this part. The optimization with ADMM for Eq. (16) includes a two-block optimization problem.

Proof. The problem for two-block convex optimization which is described as follows [37, 38]:

$$\min_{W\in\Omega_C, Y\in\Omega_Y} f(C)+g(Y) \text{ s.t. } AC+HY=Q \tag{28}$$

where Ω_C and Ω_Y denote the constraints, $f(\cdot)$ and $g(\cdot)$ denote the convex functions. A, H, Q denote vectors or matrices. The optimization for Eq.(28) can be given by

$$L(C,Y,W)=f(C)+g(Y)+\frac{\mu}{2}\left\|AC+HY-Q+\frac{W}{\mu}\right\|_F^2 \tag{29}$$

Then, the update for each variable is given as

$$\begin{array}{l} C_{t+1}=\arg\min_{C\in\Omega_C} L\left(C_t,Y_t,W_t\right) \\ Y_{t+1}=\arg\min_{Y\in\Omega_Y} L\left(C_{t+1},Y_t,W_t\right) \end{array} \tag{30}$$

$$W_{t+1}=W_t+\mu\left(AC_{t+1}+HY_{t+1}-Q\right). \tag{31}$$

It is obvious to observe that the optimization for Eq. (16) is a classical two-block optimization problem with ADMM. In particular, the update for variable $\mathbf{Z}^{(v)}$ in Eq. (16) is equal to the update for C in Eq. (30). And the update for variable $\boldsymbol{\mathcal{B}}$ is the same with the update for variable $\mathbf{Z}^{(v)}$. For the two-block optimization with ADMM, the convergent proof has been presented in [38]. Therefore, the optimization for Eq. (16) is also convergent.

Some experiments are also performed to prove the convergence property of MCGC. The error: error $=\|\boldsymbol{\mathcal{B}}-\boldsymbol{\mathcal{S}}\|_\infty$ of MCGC on COIL-20 and Scene-15 dataset in each iteration is presented in Figure 3. We can observe that along with the increase of iterations, the error values rapidly tend to 0 and remain stable which verify the convergence of MCGC experimentally. On the other datasets, the same phenomenon can also be seen.

TABLE I
STATISTICAL INFORMATION OF THE SIX BENCHMARK DATASETS.

Dataset	Samples	Type	Clusters	View dimensions
BBCSport	544	Text	5	3183/3203
COIL-20	1440	Object	20	1024/3304/6750
Flowers	1360	Flower	17	1360/1360/1360
UCI digits	2000	Digit	10	240/76/6
Scene-15	4485	Scene	15	1800/1180/1240
Caltech-101	8677	Object	101	4800/3540/1240/2048

V. EXPERIMENT RESULTS AND ANALYSES

To evaluate the effectiveness of the suggested MCGC, this section compares MCGC with other eight relevant clustering methods on six benchmark databases. All experiments were performed on Windows 10 (x64) with Intel i5-9400F (2.9 GHz) CPU and 16 GB RAM. All codes are written in Matlab 2019b.

A. Datasets

Six real public datasets are used in our experiments. And a brief summary of those datasets is presented in Table I, which include samples, type, clusters, and view dimensions. Next, we briefly describe the settings of these datasets.

BBCSport.[1] Two views with dimensions of 3, 183 and 3, 203 are adopted. **COIL-20.**[2] Similar to [30], three different features including intensity, Gabor, and LBP features are extracted. **Flowers.**[3] Similar to [27], the color feature, texture feature, and shape feature are adopted as three views. **UCI digits**[4] includes 2, 000 digit images belonging to 10 categories. Similar to [39], Fourier coefficients, pixel averaging and morphology are extracted as input features. **Scene-15** [40] includes 4, 485 images referred to 15 categories scenes. Similar to [32], three features including PHOW, LBP, and CENTRIST are adopted. **Caltech-101.** [41] Similar to [32], in addition to PHOW, LBP, and CENTRIST, the Inception V3 [42] network is also adopted.

B. Baselines and Evaluation Metrics

To comprehensively evaluate our work, our MCGC is compared with the other eight clustering methods. We perform the classic single-view clustering methods (SPCbest

[1] http://mlg.ucd.ie/datasets/segment.html
[2] http://www.cs.columbia.edu/CAVE/software/softlib/
[3] http://www.robots.ox.ac.uk/vgg/data/flowers/
[4] http://archive.ics.uci.edu/ml/datasets/Multiple+Features

[43] and LRRbest [44]) on all views and report the best results [28]. Besides, six state-of-the-art MVC methods are adopted as competitors, including RMSC [39], DiMSC [45], MCLES [46], LTMSC [30], t-SVD-MSC [31], and ETLMSC [32].

We adopt six comprehensive metrics including Normalized Mutual Information (NMI), Accuracy (ACC), Precision, F-score, Recall, and adjusted rand index (AR) to evaluate the clustering quality of these methods, in which the larger score indicate higher clustering quality. Each experiment was carried out 20 times to exclude random values, and the average values and standard deviations in parentheses are provided [28].

C. Experimental Results

The clustering results of our MCGC and all competitors are shown in Table II in which the best results are in bold, some interesting observations can be obtained.

From a global perspective, our MCGC consistently outperforms all other compared methods over all 6 metrics. For example, on the BBCSport database, MCGC improves around 7% and 17% in ACC and NMI over the sub-optimal ETLMSC method, respectively. On the Scene-15 database, MCGC outperforms the sub-optimal ETLMSC by 11% and 9% in ACC and NMI, respectively. These observations consistently validate the effectiveness and superiority of MCGC.

Next, from a local perspective, we give some fine-grained assessments of the experimental results.

- The single-view clustering methods, SPCbest [43] and LRRbest [44], are typically inferior to the multi-view ones on most databases. The reason may be that the complementary information and compatible information contained in multiple views are beneficial for the improvement of clustering performance.
- We can observe that the matrix-oriented clustering approaches (RMSC [39], DiMSC [45], and MCLES [46]) are generally inferior to the low-rank tensor-based ones (t-SVDMSC, ETLMSC, MCGC), which indicates that the low-rank property of tensor is better than that of matrix. The primary reason is that these tensor-based clustering approaches can excavate both the spatial correlation among multi-view data points as well as the view consistency information among multiple views.
- Compared with ETLMSC [32] and t-SVD-MSC [31], MCGC improves 11% and 16% in terms of ACC on Scene-15 database. Obviously, ETLMSC [32] and t-SVD-MSC [31] are worse than our MCGC on all databases over all 6 metrics. For this phenomenon, there are two main reasons. (1) The performance of MCGC is highly dependent on the learned comprehensive graph. By fully learning the similarity based on FOP and SOP among different data points, a high-quality affinity graph is achieved, which is benefit for clustering. (2) The proposed MCGC method realize the interactive learning among predefined affinity graphs and the input affinity graphs which contributes to learning the view consistency between interactive views.

D. Comparison on Affinity Graph

To verify the superiority of the learned affinity graph visually. In Figure 4, we show the contrasted affinity graphs from several approaches on the BBCSport database. As you can see, all the affinity graphs have block diagonal properties with $c = 5$ connected blocks since the BBCSport database includes 5 clusters with each block corresponds to one cluster. Meanwhile, the affinity graph of MCGC has much more clear block diagonal structure. Since a high-quality affinity with exact block diagonal structure has a good clustering performance, the above visualization results illustrate that MCGC has excellent clustering quality.

E. Comparison on Complexity

In Table III, we present the computational time and complexity on the affinity graph of the comparison methods on the Scene-15 dataset. Since all these methods use SPC on the learned affinity graph to obtain the final clustering result, which has the same complexity, we only report the computational time and complexity for learning the affinity graph. We need to mention that the number of iterations t has an obvious affect on the computational time. We can see that although the computational time of our method is longer than other comparison methods, the performance of our proposed method is obviously better than other comparison methods.

F. Sensitivity Analysis of Parameters

The sensitivity analysis of MCGC towards the hyper-parameters α and β is presented in this section. We turn α and β in the range (*i.e.*, $[10^{-2}, \cdots, 10^{5}]$) to seek the best values on different dataset. We present the NMI and ACC of MCGC versus α and β on the BBCSport and COIL-20 databases, in Figure 5 and Figure 6, respectively, it is observed that when α varies in $[10^{-2}, \cdots, 10^{3}]$ and β varies in $[10^{4}, \cdots, 10^{5}]$, our algorithm is slightly affected by α and β. This suggests that when these hyper-parameters vary across a wide range, our approach is insensitive.

VI. Conclusion

In the paper, we present the multi-view comprehensive graph clustering model. In our model, the cooperative learning based on FOP, SOP and TOP is devised to learn a comprehensive graph, which contributes to fully exploring the relation among different data points and across different views. Next, an efficient optimization scheme for solving MCGC and the theoretical guarantees are provided. In the end, we evaluate the performance of MCGC on six public databases to verify the superiority of MCGC over MVC. For future work, we wish to investigate incomplete multi-view comprehensive graph clustering, which make full use of the similarity learning based on multi-order proximity among different data points to improve clustering performance.

TABLE II
COMPARISON RESULTS ON BBCSPORT, COIL-20, FLOWERS, UCI DIGITS, SCENE-15, AND CALTECH-101 DATASET.

Datasets	Methods	NMI	ACC	Precision	F-score	Recall	AR
BBCSport	SPCbest [43]	0.344(0.012)	0.036(0.002)	0.369(0.020)	0.249(0.001)	0.297(0.007)	0.381(0.005)
	LRRbest [44]	0.698(0.002)	0.836(0.001)	0.768(0.001)	0.776(0.001)	0.784(0.001)	0.705(0.001)
	RMSC [39]	0.666(0.001)	0.826(0.001)	0.766(0.001)	0.719(0.001)	0.677(0.001)	0.637(0.001)
	DiMSC [45]	0.785(0.000)	0.922(0.000)	0.846(0.000)	0.858(0.000)	0.872(0.000)	0.813(0.000)
	MCLES [46]	0.802(0.000)	0.921(0.000)	0.827(0.000)	0.845(0.000)	0.865(0.000)	0.795(0.000)
	LTMSC [30]	0.230(0.018)	0.476(0.030)	0.335(0.020)	0.432(0.010)	0.335(0.020)	0.178(0.031)
	t-SVD-MSC [31]	0.894(0.000)	0.949(0.000)	0.935(0.000)	0.938(0.000)	0.940(0.000)	0.918(0.000)
	ETLMSC [32]	0.827(0.000)	0.934(0.000)	0.901(0.000)	0.877(0.000)	0.855(0.000)	0.840(0.000)
	MCGC	**1.000(0.000)**	**1.000(0.000)**	**1.000(0.000)**	**1.000(0.000)**	**1.000(0.000)**	**1.000(0.000)**
COIL-20	SPCbest [43]	0.806(0.008)	0.672(0.063)	0.596(0.021)	0.640(0.017)	0.692(0.013)	0.619(0.018)
	LRRbest [44]	0.829(0.006)	0.761(0.003)	0.717(0.003)	0.734(0.006)	0.751(0.002)	0.720(0.020)
	RMSC [39]	0.800(0.017)	0.685(0.045)	0.620(0.057)	0.656(0.042)	0.698(0.026)	0.637(0.044)
	DiMSC [45]	0.846(0.002)	0.778(0.022)	0.739(0.007)	0.745(0.005)	0.751(0.003)	0.732(0.005)
	MCLES [46]	0.740(0.019)	0.706(0.026)	0.505(0.036)	0.553(0.029)	0.611(0.021)	0.521(0.032)
	LTMSC [30]	0.860(0.002)	0.804(0.011)	0.741(0.009)	0.760(0.007)	0.776(0.006)	0.748(0.004)
	t-SVD-MSC [31]	0.884(0.005)	0.830(0.000)	0.785(0.007)	0.800(0.004)	0.808(0.001)	0.786(0.003)
	ETLMSC [32]	0.945(0.030)	0.873(0.065)	0.840(0.068)	0.872(0.060)	0.907(0.055)	0.865(0.063)
	MCGC	**0.997(0.009)**	**0.992(0.024)**	**0.990(0.031)**	**0.992(0.024)**	**0.995(0.017)**	**0.992(0.025)**
Flowers	SPCbest [43]	0.027(0.002)	0.070(0.003)	0.932(0.140)	0.110(0.002)	0.058(0.000)	0.001(0.000)
	LRRbest [44]	0.419(0.006)	0.396(0.009)	0.279(0.004)	0.284(0.003)	0.290(0.003)	0.239(0.003)
	RMSC [39]	0.396(0.001)	0.385(0.016)	0.234(0.012)	0.249(0.011)	0.256(0.010)	0.231(0.019)
	DiMSC [45]	0.442(0.011)	0.434(0.014)	0.302(0.007)	0.310(0.008)	0.318(0.010)	0.266(0.009)
	MCLES [46]	0.516(0.000)	0.469(0.000)	0.342(0.000)	0.390(0.000)	0.462(0.000)	0.337(0.000)
	LTMSC [46]	0.478(0.008)	0.476(0.012)	0.347(0.009)	0.354(0.008)	0.361(0.008)	0.313(0.009)
	t-SVD-MSC [31]	0.852(0.002)	0.836(0.005)	0.772(0.002)	0.780(0.002)	0.789(0.002)	0.766(0.002)
	ETLMSC [32]	0.874(0.025)	0.811(0.066)	0.748(0.064)	0.778(0.054)	0.810(0.041)	0.763(0.057)
	MCGC	**0.952(0.009)**	**0.951(0.015)**	**0.914(0.021)**	**0.919(0.019)**	**0.924(0.017)**	**0.914(0.020)**
UCI digits	SPCbest [43]	0.642(0.021)	0.731(0.034)	0.582(0.030)	0.591(0.029)	0.601(0.030)	0.545(0.033)
	LRRbest [44]	0.743(0.021)	0.720(0.047)	0.734(0.021)	0.690(0.028)	0.652(0.033)	0.654(0.032)
	RMSC [39]	0.822(0.026)	0.915(0.036)	0.797(0.065)	0.811(0.049)	0.826(0.031)	0.789(0.055)
	DiMSC [45]	0.782(0.002)	0.867(0.001)	0.769(0.002)	0.772(0.002)	0.775(0.002)	0.747(0.002)
	MCLES [46]	0.891(0.008)	0.941(0.004)	0.885(0.008)	0.889(0.008)	0.894(0.007)	0.877(0.009)
	LTMSC [30]	0.762(0.009)	0.792(0.009)	0.724(0.012)	0.737(0.013)	0.749(0.013)	0.707(0.014)
	t-SVD-MSC [31]	0.934(0.001)	0.966(0.001)	0.933(0.001)	0.935(0.001)	0.936(0.001)	0.928(0.001)
	ETLMSC [32]	0.970(0.013)	0.941(0.023)	0.935(0.031)	0.936(0.027)	0.938(0.024)	0.933(0.029)
	MCGC	**0.999(0.000)**	**0.999(0.000)**	**0.999(0.000)**	**0.999(0.000)**	**0.999(0.000)**	**0.999(0.000)**
Scene-15	SPCbest [43]	0.421(0.010)	0.437(0.015)	0.314(0.016)	0.321(0.022)	0.329(0.020)	0.270(0.010)
	LRRbest [44]	0.426(0.018)	0.445(0.013)	0.316(0.015)	0.324(0.010)	0.333(0.015)	0.272(0.015)
	RMSC [39]	0.564(0.023)	0.507(0.017)	0.425(0.021)	0.437(0.019)	0.450(0.024)	0.394(0.025)
	DiMSC [45]	0.269(0.009)	0.300(0.010)	0.173(0.016)	0.181(0.012)	0.190(0.010)	0.117(0.012)
	LTMSC [30]	0.571(0.011)	0.574(0.009)	0.452(0.003)	0.465(0.007)	0.479(0.008)	0.424(0.010)
	t-SVD-MSC [31]	0.858(0.007)	0.812(0.007)	0.743(0.006)	0.788(0.001)	0.839(0.003)	0.771(0.003)
	ETLMSC [32]	0.879(0.006)	0.865(0.030)	0.825(0.020)	0.828(0.018)	0.832(0.016)	0.816(0.019)
	MCGC	**0.962(0.009)**	**0.972(0.024)**	**0.959(0.025)**	**0.957(0.021)**	**0.955(0.016)**	**0.954(0.022)**
Caltech-101	SPCbest [43]	0.723(0.032)	0.484(0.019)	0.597(0.018)	0.340(0.025)	0.235(0.020)	0.319(0.014)
	LRRbest [44]	0.728(0.014)	0.510(0.009)	0.627(0.012)	0.339(0.008)	0.231(0.010)	0.304(0.017)
	RMSC [39]	0.573(0.047)	0.346(0.036)	0.457(0.033)	0.258(0.027)	0.182(0.031)	0.246(0.031)
	DiMSC [45]	0.589(0.011)	0.351(0.008)	0.362(0.010)	0.253(0.007)	0.191(0.007)	0.226(0.003)
	LTMSC [30]	0.788(0.005)	0.559(0.012)	0.670(0.009)	0.403(0.003)	0.288(0.012)	0.393(0.007)
	t-SVD-MSC [31]	0.858(0.003)	0.607(0.005)	0.742(0.007)	0.440(0.010)	0.323(0.009)	0.430(0.005)
	ETLMSC [32]	0.898(0.006)	0.593(0.025)	0.779(0.029)	0.452(0.019)	0.319(0.002)	0.443(0.019)
	MCGC	**0.903(0.003)**	**0.663(0.011)**	**0.817(0.014)**	**0.469(0.011)**	**0.329(0.009)**	**0.461(0.011)**

REFERENCES

[1] R. Vidal and P. Favaro, “Low rank subspace clustering,” *Pattern Recognition Letters*, vol. 43, pp. 47–61, 2014.

[2] J. Ma, Y. Zhang, and L. Zhang, “Discriminative subspace matrix factorization for multiview data clustering,” *Pattern Recognition*, vol. 111, p. 107676, 2021.

[3] S. N. Sulaiman and N. A. Mat Isa, “Adaptive fuzzy-k-means clustering algorithm for image segmentation,” *IEEE Transactions on Consumer Electronics*, vol. 56, no. 4, pp. 2661–2668, 2010.

[4] Q. Liu, K. M. Kamoto, X. Liu, M. Sun, and N. Linge, “Low-complexity non-intrusive load monitoring using unsupervised learning and generalized appliance models,” *IEEE Transactions on Consumer Electronics*, vol. 65, no. 1, pp. 28–37, 2019.

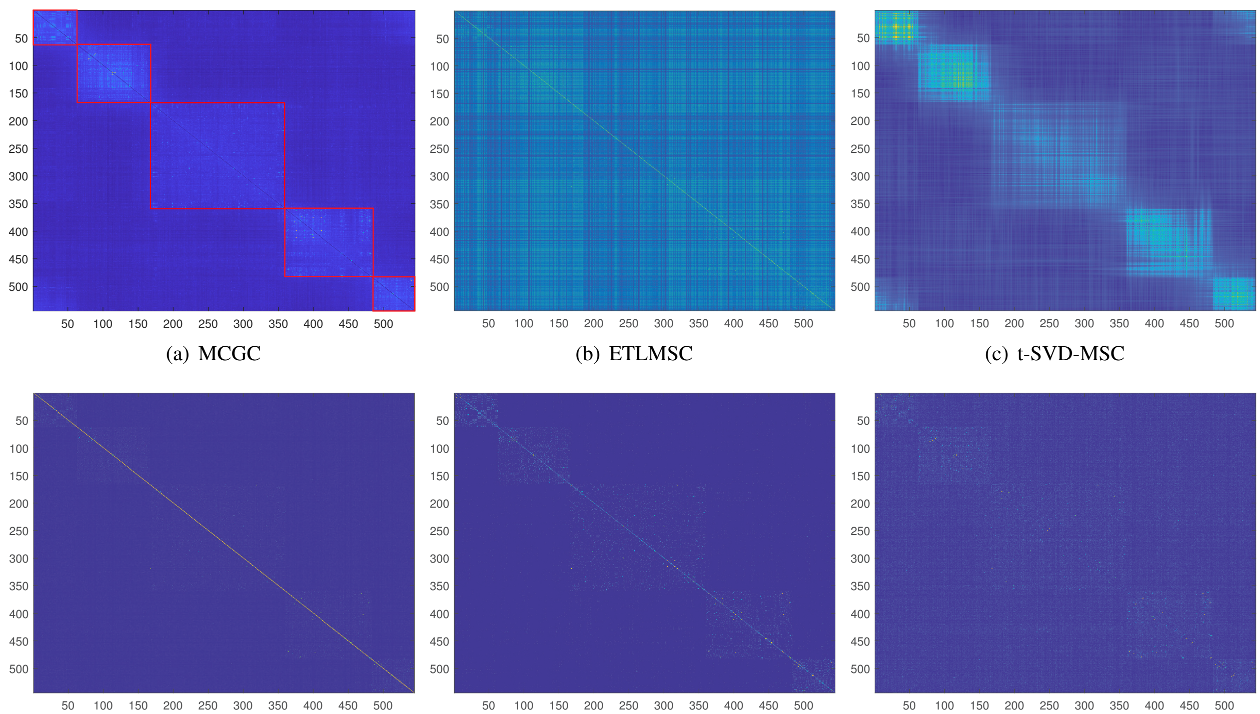
[5] J. Baek, S. K. An, and P. Fisher, “Dynamic cluster header selection and conditional re-clustering for wireless sensor networks,” *IEEE Transactions on Consumer Electronics*, vol. 56, no. 4, pp. 2249–2257, 2010.

[6] M. Saha, “A graph based approach to multiview clustering,” in *Proceedings of Pattern Recognition and Machine*

TABLE III
COMPUTATIONAL TIME AND COMPLEXITY ON THE AFFINITY GRAPHS OF THE COMPARISON METHODS ON THE SCENE-15 DATASET.

Method	Time(s)	Time Complexity
LRRbest [44]	0.62	$O\left(t\left(d^2N+d^3\right)\right),(d\leq N)$
RMSC [39]	943.97	$\mathcal{O}\left(tN^3\right)$
DiMSC [45]	1123.25	$\mathcal{O}\left(tVN^3\right)$
LTMSC [30]	4186.55	$\mathcal{O}\left(tVN^3\right)$
t-SVD-MSC [31]	1493.36	$\mathcal{O}\left(t(2N^2V\log(N)+N^2V^2)\right)$
ETLMSC [32]	229.54	$\mathcal{O}\left(tVN^2\log(N)\right)$
MCGC	2333.19	$\mathcal{O}\left(t\left(VN^2+VN^2\log(N)+V^2N^2+V\right)\right)$

Note: All these methods use SPC on the learned graph to obtain the final clustering results, which have the same computational time in different methods, therefore, we only report the computational time and complexity on the affinity graph learning.



(a) MCGC (b) ETLMSC (c) t-SVD-MSC
(d) LTMSC (e) MCLES (f) DiMSC

Fig. 4. Visualizations for the learned affinity graphs obtained from the contrast approaches on the BBCSport database. As presented, the affinity graph learned by MCGC has much more clear structure.

Intelligence 5th International Conference. Springer, 2013, pp. 128–133.

[7] P. Chen, L. Liu, Z. Ma, and Z. Kang, “Smoothed multi-view subspace clustering,” in *Proceedings of the Neural Computing for Advanced Applications*. Springer, 2021, pp. 128–140.

[8] J. Gong, N. Liao, C. Li, X. Ma, W. He, and B. Guo, “Superpixel segmentation via contour optimized non-iterative clustering,” in *Proceedings of the Neural Computing for Advanced Applications*. Springer, 2021, pp. 645–658.

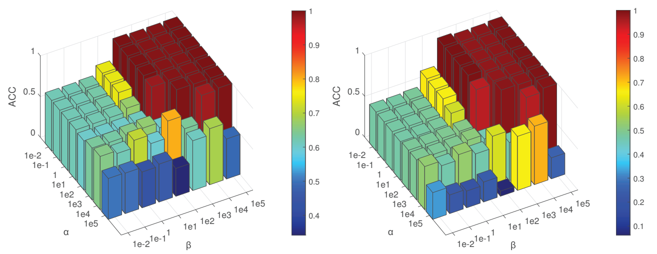
[9] E. Elhamifar and R. Vidal, “Sparse subspace clustering: Algorithm, theory, and applications,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2765–2781, 2013.

[10] J. Zhao, X. Xie, X. Xu, and S. Sun, “Multi-view learning overview: Recent progress and new challenges,” *Information Fusion*, vol. 38, pp. 43–54, 2017.

[11] F. Nie, X. Wang, and H. Huang, “Clustering and projected clustering with adaptive neighbors,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2014, pp. 977–986.

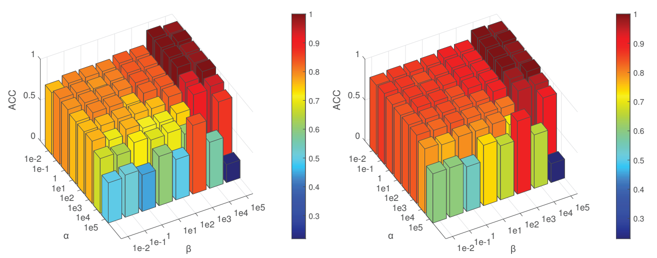
[12] E. Elhamifar and R. Vidal, “Clustering disjoint subspaces via sparse representation,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, 2010, pp. 1926–1929.

[13] H. Wang, Y. Yang, and B. Liu, “Gmc: Graph-based multi-



(a) The ACC versus α and β (b) The NMI versus α and β

Fig. 5. The ACC and NMI of MCGC on the BBCSport database with different α and β.



(a) The ACC versus α and β (b) The NMI versus α and β

Fig. 6. The ACC and NMI of MCGC on the COIL20 database with different α and β.

view clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 6, pp. 1116–1129, 2020.

[14] Y. Mei, Z. Ren, B. Wu, Y. Shao, and T. Yang, "Robust graph-based multi-view clustering in latent embedding space," *International Journal of Machine Learning and Cybernetics*, vol. 13, pp. 497–508, 2022.

[15] X. Wang, Y. Lu, C. Shi, R. Wang, P. Cui, and S. Mou, "Dynamic heterogeneous information network embedding with meta-path based proximity," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 3, pp. 1117–1132, 2022.

[16] X. Zhang, H. Sun, Z. Liu, Z. Ren, Q. Cui, and Y. Li, "Robust low-rank kernel multi-view subspace clustering based on the schatten p-norm and correntropy," *Information Sciences*, vol. 477, pp. 430–447, 2019.

[17] D. Y. Xie, W. Xia, Q. Wang, Q. Gao, and S. Xiao, "Multi-view clustering by joint manifold learning and tensor nuclear norm," *Neurocomputing*, vol. 380, pp. 105–114, 2020.

[18] H. Wang, G. Han, J. Li, B. Zhang, J. Chen, Y. Hu, C. Han, and H. Cai, "Learning task-driving affinity matrix for accurate multi-view clustering through tensor subspace learning," *Information Sciences*, vol. 563, pp. 290–308, 2021.

[19] Z. Ren, M. Mukherjee, M. Bennis, and J. Lloret, "Multikernel clustering via non-negative matrix factorization tailored graph tensor over distributed networks," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 7, pp. 1946–1956, 2021.

[20] Y. Chen, X. Xiao, and Y. Zhou, "Multi-view subspace clustering via simultaneously learning the representation tensor and affinity matrix," *Pattern Recognition*, vol. 106, p. 107441, 2020.

[21] Z. Ren, H. Li, C. Yang, and Q. Sun, "Multiple kernel subspace clustering with local structural graph and low-rank consensus kernel learning," *Knowledge Based System*, vol. 188, p. 105040, 2020.

[22] Z. Kang, Y. Lu, Y. Su, C. Li, and Z. Xu, "Similarity learning via kernel preserving embedding," in *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*. AAAI Press, 2019, pp. 4057–4064.

[23] X. Zhang, H. Sun, Z. Liu, Z. Ren, Q. Cui, and Y. Li, "Robust low-rank kernel multi-view subspace clustering based on the schatten *p*-norm and correntropy," *Information Sciences*, vol. 477, pp. 430–447, 2019.

[24] F. Nie, J. Li, and X. Li, "Parameter-free auto-weighted multiple graph learning: A framework for multiview clustering and semi-supervised classification," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. IJCAI/AAAI Press, 2016, pp. 1881–1887.

[25] C. Yang, Z. Ren, Q. Sun, M. Wu, M. Yin, and Y. Sun, "Joint correntropy metric weighting and block diagonal regularizer for robust multiple kernel subspace clustering," *Information Sciences*, vol. 500, pp. 48–66, 2019.

[26] G. F. Lu, H. Li, Y. Wang, and G. Tang, "Multi-view subspace clustering with kronecker-basis-representation-based tensor sparsity measure," *Machine vision and applications*, vol. 32, no. 6, pp. 1–12, 2021.

[27] Y. Chen, S. Wang, C. Peng, Z. Hua, and Y. Zhou, "Generalized nonconvex low-rank tensor approximation for multi-view subspace clustering," *IEEE Transactions on Image Processing*, vol. 30, pp. 4022–4035, 2021.

[28] Y. Mei, Z. Ren, B. Wu, T. Yang, and Y. Shao, "Robust kernelized multiview clustering based on high-order similarity learning," *IEEE Access*, vol. 10, pp. 54 221–54 234, 2022.

[29] J. Guo, Y. Sun, J. Gao, Y. Hu, and B. Yin, "Logarithmic schatten-p norm minimization for tensorial multi-view subspace clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2022.

[30] C. Zhang, H. Fu, S. Liu, G. Liu, and X. Cao, "Low-rank tensor constrained multiview subspace clustering," in *Proceedings of the 2015 IEEE International Conference on Computer Vision*. IEEE Computer Society, 2015, pp. 1582–1590.

[31] Y. Xie, D. Tao, W. Zhang, Y. Liu, L. Zhang, and Y. Qu, "On unifying multi-view self-representations for clustering by tensor multi-rank minimization," *International Journal of Computer Vision*, vol. 126, p. 1157–1179, 11 2018.

[32] J. Wu, Z. Lin, and H. Zha, "Essential tensor learning for multi-view spectral clustering," *IEEE Transactions on Image Processing*, vol. 28, no. 12, pp. 5910–5922, 2019.

[33] M. Chen, C. Wang, and J. Lai, "Low-rank tensor based proximity learning for multi-view clustering," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2022.

[34] P. Goyal and E. Ferrara, "Graph embedding techniques, applications, and performance: A survey," *Knowledge Based System*, vol. 151, pp. 78–94, 2018.

[35] Z. Ren, Q. Sun, and D. Wei, "Multiple kernel clustering with kernel k-means coupled graph tensor learning," in *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence*. AAAI Press, 2021, pp. 9411–9418.

[36] F. Nie, X. Wang, M. Jordan, and H. Huang, "The constrained laplacian rank algorithm for graph-based clustering," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. AAAI Press, 2016, pp. 1969–1976.

[37] R. Liu, Z. Lin, and Z. Su, "Linearized alternating direction method with parallel splitting and adaptive penalty for separable convex programs in machine learning," *Machine Learning*, vol. 99, pp. 287–325, 10 2013.

[38] E. Esser, "Applications of lagrangian-based alternating direction methods and connections to split bregman," *CAM Rep*, vol. 9, pp. 1–32, 01 2009.

[39] R. Xia, Y. Pan, L. Du, and J. Yin, "Robust multi-view spectral clustering via low-rank and sparse decomposition," in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*. AAAI Press, 2014, pp. 2149–2155.

[40] F. Li and P. Pietro, "A bayesian hierarchical model for learning natural scene categories," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2005, pp. 524–531.

[41] F. Li, F. Robert, and P. Pietro, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," *Computer Vision and Image Understanding*, vol. 106, no. 11, pp. 59–70, 2007.

[42] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2016, pp. 2818–2826.

[43] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proceedings of the Neural Information Processing Systems: Natural and Synthetic (NIPS)*, vol. 2, 2002, pp. 849–856.

[44] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 171–184, 2013.

[45] X. Cao, C. Zhang, H. Fu, S. Liu, and H. Zhang, "Diversity-induced multi-view subspace clustering," in *Proceedings of the 2015 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2015, pp. 586–594.

[46] M. Chen, L. Huang, C. Wang, and D. Huang, "Multi-view clustering in latent embedding space," in *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*. AAAI Press, 2020, pp. 3513–3520.

Yanying Mei received the B.E. Degree from North China University of Water Resources and Electric Power, Zhengzhou, China, in 2012, and M.S. Degree from Southwest University of Science and Technology, Mianyang, China, in 2015, respectively. She is currently pursuing the Ph.D. Degree. Her research interests include machine learning, high-dimensional statistics and clustering.

Zhenwen Ren (Member, IEEE) received the Ph.D. degrees in the Control Science and Engineering with the Nanjing University of Science and Technology (NJUST), Nanjing, China, in 2021. He is working with the school of National Defence Science and Technology, Southwest University of Science and Technology (SWUST). He has published 60+ peer-reviewed papers, including those in highly regarded conferences and journals, such as the AAAI, the ACM Multimedia, the IEEE Transactions on Neural Networks and Learning Systems, the IEEE Transactions on Cybernetics, the IEEE Transactions on Image Processing, the IEEE Transactions on Knowledge and Data Engineering, the IEEE Transactions on Industrial Informatics, and the Pattern Recognition. His research interests include computer vision, machine learning, deep learning, and industrial software.

Bin Wu is a professor in Southwest University of Science and Technology, He received the B.E. Degree in Central South University, Changsha, China, in 1985, the M.S. and Ph.D. Degree in University of Science and Technology Beijing, Beijing, China, in 1993 and 1999. His research include the most economical intelligent control, image processing.

Tao Yang is a professor in Southwest University of Science and Technology, He received the B.E. degree from Xi'an Jiaotong University, Xian, China, in 1993, the M.E. degree from the South China University of Technology, Guangzhou, China, in 1996, and the Ph.D. degree from Tsinghua University, Beijing, China, in 2003, respectively. His current research interests are in signal processing, and modeling, simulation and control of electromechanical systems, including acoustic array signal processing for robot scene perception and machine fault diagnosis, and micro electromechanical systems.

Yanhua Shao received an MS degree in pattern recognition and intelligent system from Southwest University of Science and Technology, China, in 2010, the Ph.D. Degree in instrument science and technology at Chongqing University. His research include machine learning and computer vision.