Hierarchical Representation for Multi-view Clustering: from Intra-sample to Intra-view to Inter-view

Jing-Hua Yang
Macau University of Science and Technology
Macau, China
yangjinghua110@126.com

Chuan Chen*
Sun Yat-Sen University
Guangzhou, China
chenchuan@mail.sysu.edu.cn

Hong-Ning Dai
Hong Kong Baptist University
Hong Kong, China
hndai@ieee.org

Meng Ding
Southwest Jiaotong University
Chengdu, China
dingmeng56@163.com

Le-Le Fu
Sun Yat-Sen University
Guangzhou, China
lawrencefzu@gmail.com

Zibin Zheng
Sun Yat-Sen University
Guangzhou, China
zhzibin@mail.sysu.edu.cn

ABSTRACT

Multi-view clustering (MVC) aims at exploiting the consistent features within different views to divide samples into different clusters. Existing subspace-based MVC algorithms usually assume linear subspace structures and two-stage similarity matrix construction strategies, thereby posing challenges in imprecise low-dimensional subspace representation and inadequacy of exploring consistency. This paper presents a novel hierarchical representation for MVC method via the integration of intra-sample, intra-view, and inter-view representation learning models. In particular, we first adopt the deep autoencoder to adaptively map the original high-dimensional data into the latent low-dimensional representation of each sample. Second, we use the self-expression of the latent representation to explore the global similarity between samples of each view and obtain the subspace representation coefficients. Third, we construct the third-order tensor by arranging multiple subspace representation matrices and impose the tensor low-rank constraint to sufficiently explore the consistency among views. Being incorporated into a unified framework, these three models boost each other to achieve a satisfactory clustering result. Moreover, an alternating direction method of multipliers algorithm is developed to solve the challenging optimization problem. Extensive experiments on both simulated and real-world multi-view datasets show the superiority of the proposed method over eight state-of-the-art baselines.

CCS CONCEPTS

• **Information systems → Clustering**; • **Computing methodologies → Cluster analysis**.

*Corresponding author.

KEYWORDS

Multi-view clustering, hierarchical representation, tensor low-rank constraint, deep autoencoder, alternating direction method of multipliers

1 INTRODUCTION

With the rapid development of information acquisition technology, the collected big data often exhibit *multi-view* characteristics in terms of different views or sources [6, 15, 32, 39, 46]. For example, the published news may be reported and described by videos or texts while a text description can be expressed in Chinese, English, or other languages. The multi-view data usually admit abundant potential consistency and complementarity, which can be extracted by multi-view learning. Among multi-view learning approaches, multi-view clustering (MVC) methods learn the consensus features within different views and divide data samples into corresponding disjoint clusters without ground truth [11, 23, 33, 34, 48].

Generally, the underlying clusters may be masked due to high dimension and redundancy of the observed data. To remove irrelevant or redundant dimension, *subspace clustering* has received extensive attention in data mining and machine learning [10, 26, 50, 56]. The goal of subspace clustering is to find a compact low-dimensional subspace suitable for each group of data samples and divide them into multiple clusters. These methods generally obtain clustering results in three steps. First, the representation coefficients are obtained by solving the optimization problem of different subspace representations, such as the sparse representation [10] and the low-rank representation [18, 26]. Then, an affinity matrix is constructed according to the representation coefficients. Finally, the clustering results are obtained by using spectral clustering tools [36] on the affinity matrix. Following this idea, there are a number of subspace clustering studies, such as [2, 25, 27, 37]. Despite the advent of these subspace clustering methods on single-view data, they are not suitable for processing multi-view data. To explore high-order
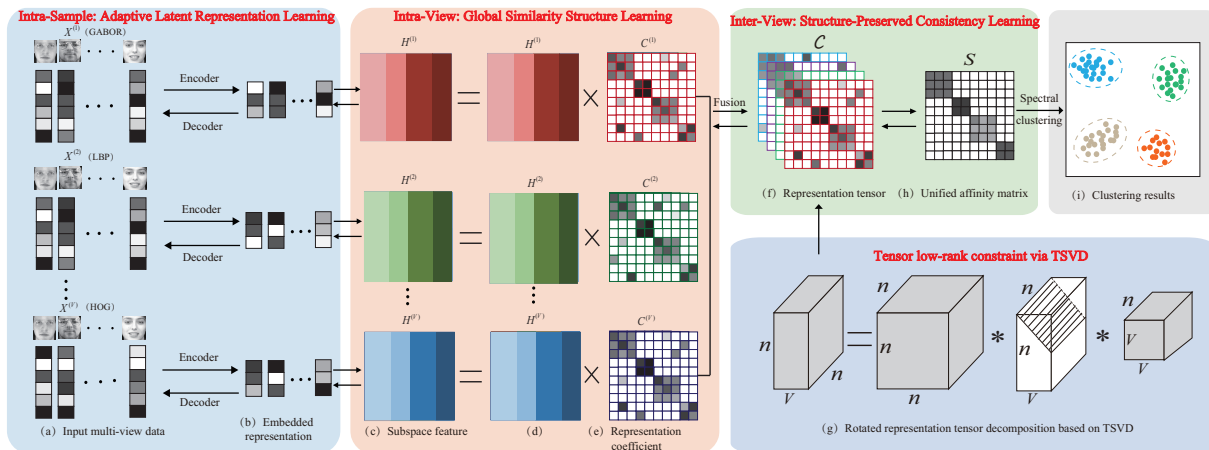
Figure 1: Overview of HRMC. For multi-view data (a), we first use deep autoencoder to explore the latent nonlinear low-dimensional representation (b) of each sample. Second, based on self-expression, we construct the representation coefficient (e) of the feature matrix (d) to guarantee the global similar structure between samples in each view. Third, we fuse representations of all views as a third-order tensor (f) and introduce the tensor low-rank constraint on the rotated tensor (g) to capture high-order correlation among multiple views. Fourth, the representation (f) and the affinity matrix (h) are simultaneously optimized to strengthen each other. Finally, we obtain clustering results (i) after applying spectral clustering on the affinity matrix.

correlations among multiple views, many tensor-based multi-view subspace clustering methods were proposed. For instance, Xie *et al.* [42] used the tensor nuclear norm based on the tensor-singular value decomposition, and Gao *et al.* [12] further used the weighted tensor nuclear norm for multi-view clustering. Differently, Chen *et al.* [3] performed multi-view clustering on the latent embedding space instead of the original feature space. Moreover, Xiao *et al.* [40] used the prior knowledge and the tensor low-rank constraint to learn the self-representation tensor.

Although the aforementioned methods achieve competitive performance, they still have the following limitations. (1) They assume that the multi-view data admit the *linear* subspace structure though the subspace structure is usually *nonlinear* and even complex—this may lead to the failure of finding the compact latent low-dimensional subspace. (2) They usually *separately* construct two key components in MVC: the affinity matrix and the low-rank representation tensor, but the close dependence between them is ignored, thereby resulting in the inadequacy of exploring consistency. To address these two limitations, some studies such as [5, 47, 54] employed the kernel-based mapping function to explore the nonlinear subspace structure while other studies like [4, 5, 40] proposed to jointly optimize the representation tensor and the affinity matrix. However, the kernel selection is a difficult task and unsuitable kernel functions may lead to the unsatisfactory clustering performance [16, 29]. Moreover, all these methods cannot fully extract the hierarchical and nonlinear structure in multi-view data since they are restricted to *shallow* architectures.

Recently, owing to the powerful ability of deep learning, deep multi-view clustering methods have been proposed to learn the hierarchical structure embedded in multi-view data [13, 17, 19, 24, 41, 55]. Taking the advantage of deep representation, Wang *et al.* [38] proposed a deep model, which integrates autoencoders and

canonical correlation analysis (CCA) to capture the structure information. However, CCA is only suitable for handling the two-view case and is incapable of handling more views though the number of views in real datasets is often more than two [28]. Moreover, Zhao *et al.* [52] used deep matrix factorization to learn the consensus latent representation for MVC. Zhang *et al.* [49] proposed a deep clustering method using mixture of autoencoders. Du *et al.* [9] adopted multiple aotoencoders to learn the nonlinear structure of each view and explored the consistency between any two views together. Although these methods perform well for capturing the unique information within each view, they cannot fully explore the consistent and complementary information across different views.

In this work, we present a Hierarchical Representation-based Multi-view Clustering (HRMC). In particular, the proposed HRMC framework consists of intra-sample adaptive latent representation learning, intra-view global similarity structure learning, and inter-view structure-preserved consistency learning, as shown in Fig. 1. HRMC is briefly described as follows:

(1) **At the intra-sample adaptive latent representation learning**, we adopt deep autoencoders to adaptively map the original high-dimensional data into the latent low-dimensional representation. This manner not only captures the nonlinear structure within each sample but also preserves the local details of original data as many as possible.

(2) **At the intra-view global similarity structure learning**, inspired by the self-expression subspace clustering, we use the self-expression strategy to learn the representation coefficient of the latent low-dimensional representation, thereby exploring global correlations among samples within each view.

(3) **At the inter-view structure-preserved consistency learning**, we apply the tensor nuclear norm constraint on the third-order tensor constructed by all representation coefficients to learn the

unified similarity matrix. Constructing a third-order tensor can preserve the overall structure of representation coefficients—thereby better exploring consistency information among multiple views.

The proposed hierarchical representation learning not only explores the view-specific local and global structures but also fully captures the high-order correlations and consistency among multiple views. The main contributions of this paper are listed as follows:

• We propose a hierarchical representation learning-based method to obtain the consistency information among different views, thereby simultaneously exploring the complex nonlinear structure hidden in each sample, the global correlation of the latent representations in each view, and the high-order correlation within all representation coefficients, to cluster multi-view data (Sec. 3).

• We integrate these three representation learning models into a unified framework. Therefore, the latent embedding representations, self-expression coefficients, and the uniform similarity matrix are *jointly* optimized to yield a satisfactory clustering result.

• An efficient algorithm is developed to solve the proposed optimization problem by the alternating direction method of multipliers (Sec. 4). Extensive experiments on both simulated and real-world datasets show the effectiveness and superiority of the proposed method (Sec. 5). The ablation experiments also verify the benefits of three hierarchical representations to the proposed method (Sec. 6).

2 NOTATIONS AND PRELIMINARIES

2.1 Notations

We use the calligraphy letter $\mathcal{Z}$, the upper case letter Z, the bold lower case letter $\mathbf{z}$, the lower case letter z to respectively denote the tensor, the matrix, the vector, and the scalar. The (i, j)-th element of $Z \in \mathbb{R}^{n_1 \times n_2}$ and (i, j, k)-th element of $\mathcal{Z} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ are denoted by $Z_{i,j}$ and $\mathcal{Z}_{i,j,k}$, respectively. For a third-order tensor $\mathcal{Z}$, we use $\mathcal{Z}(i,:,:)$, $\mathcal{Z}(:,j,:)$, and $\mathcal{Z}(:,:,k)$ (or $\mathcal{Z}^{(k)}$) to denote the i-th horizontal, j-th lateral, and k-th frontal slices, respectively. The infinity norm of $\mathcal{Z}$ is $\|\mathcal{Z}\|_\infty = \max_{i,j,k} |\mathcal{Z}_{i,j,k}|$. The Frobenius norms of Z and $\mathcal{Z}$ are denoted by $\|Z\|_F = \sqrt{\sum_{i,j} Z_{i,j}^2}$ and $\|\mathcal{Z}\|_F = \sqrt{\sum_{i,j,k} \mathcal{Z}_{i,j,k}^2}$, respectively.

2.2 Preliminaries

For $\mathcal{Z} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, we denote the discrete Fourier transform (DFT) of $\mathcal{Z}$ along the third dimension by $\bar{\mathcal{Z}}$, i.e., $\bar{\mathcal{Z}} = \text{fft}(\mathcal{Z}, [], 3)$, and we can obtain $\mathcal{Z}$ by inverse DFT, i.e., $\mathcal{Z} = \text{ifft}(\bar{\mathcal{Z}}, [], 3)$. The block diagonal matrix of $\mathcal{Z}$ is

$$\text{bdiag}(\mathcal{Z}) = \begin{pmatrix} \mathcal{Z}^{(1)} & & & \\ & \mathcal{Z}^{(2)} & & \\ & & \ddots & \\ & & & \mathcal{Z}^{(n_3)} \end{pmatrix}. \quad (1)$$

The block-circulant matrix of $\mathcal{Z}$ has the following form:

$$\text{bcirc}(\mathcal{Z}) = \begin{pmatrix} \mathcal{Z}^{(1)} & \mathcal{Z}^{(n_3)} & \cdots & \mathcal{Z}^{(2)} \\ \mathcal{Z}^{(2)} & \mathcal{Z}^{(1)} & \cdots & \mathcal{Z}^{(3)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{Z}^{(n_3)} & \mathcal{Z}^{(n_3-1)} & \cdots & \mathcal{Z}^{(1)} \end{pmatrix}. \quad (2)$$

In addition, the unfold and fold operators of $\mathcal{Z}$ are defined as

$$\text{unfold}(\mathcal{Z}) = \left(\mathcal{Z}^{(1)}; \mathcal{Z}^{(2)}; \ldots; \mathcal{Z}^{(n_3)} \right), \text{ fold}(\text{unfold}(\mathcal{Z})) = \mathcal{Z}. \quad (3)$$

To better understand the definition of tensor nuclear norm used in our work, we first give some related definitions.

DEFINITION 1. *(**Tensor-tensor product**) The tensor-tensor product of $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{B} \in \mathbb{R}^{n_2 \times l \times n_3}$ is*

$$C = \mathcal{A} * \mathcal{B} = \text{fold}(\text{bcirc}(\mathcal{A}) \times \text{unfold}(\mathcal{B})),$$

where $C \in \mathbb{R}^{n_1 \times l \times n_3}$ and $\times$ denotes the matrix product.

DEFINITION 2. *(**Transpose tensor**) For a tensor $\mathcal{Z} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the transpose tensor $\mathcal{Z}^\top \in \mathbb{R}^{n_2 \times n_1 \times n_3}$ can be obtained by firstly transposing each of frontal slices and then reversing the order of transposed frontal slices 2 through n_3.*

DEFINITION 3. *(**Identity tensor**) For the identity tensor $I \in \mathbb{R}^{n_1 \times n_1 \times n_3}$, whose the first frontal slice is an $n_1 \times n_1$ identity matrix and whose other frontal slices are all zeros.*

DEFINITION 4. *(**Orthogonal tensor**) A tensor $\mathcal{Z}$ is orthogonal if it satisfies $\mathcal{Z}^\top * \mathcal{Z} = \mathcal{Z} * \mathcal{Z}^\top = I$.*

DEFINITION 5. *(**Tensor Singular Value Decomposition (TSVD)**) Let $\mathcal{Z} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, then $\mathcal{Z}$ has TSVD as*

$$\mathcal{Z} = \mathcal{U} * S * \mathcal{V}^\top, \quad (4)$$

where $\mathcal{U} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$ and $\mathcal{V} \in \mathbb{R}^{n_2 \times n_2 \times n_3}$ are orthogonal tensors, and $S \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is a diagonal tensor.

DEFINITION 6. *(**Tensor Nuclear Norm (TNN)**) Given a third-order tensor $\mathcal{Z} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, TNN is defined as*

$$\|\mathcal{Z}\|_* = \sum_{k=1}^{n_3} \|\bar{\mathcal{Z}}^{(k)}\|_*, \quad (5)$$

where $\|\bar{\mathcal{Z}}^{(k)}\|_*$ is the nuclear norm of $\bar{\mathcal{Z}}^{(k)}$.

3 THE PROPOSED MODEL

This section details the proposed model. We first introduce some symbols. Given a multi-view dataset $\widetilde{X} = \{X^{(v)} \in \mathbb{R}^{d^{(v)} \times N}\}_{v=1}^V$, where $X^{(v)}$ denotes the feature matrix of the v-th view, $d^{(v)}$ denotes the dimension of the v-th feature space, N is the number of samples, and V is the total number of views. We then elaborate on three hierarchical representation learning models: adaptive latent representation learning (Sec. 3.1), global similarity structure learning (Sec. 3.2), and structure-preserved consistency learning (Sec. 3.3).

3.1 Intra-sample Adaptive Latent Representation Learning

The practical multi-view data usually admits high dimension and complex linear or nonlinear structure information. Therefore, one effective way is to find a mapping that transforms the input data to a low-dimensional latent representation, which maintains the local information of the input data.

We adaptively generate the low-dimensional latent representation in the framework of deep autoencoder [31, 45], to capture the

complex structure information of each sample. The deep autoencoder contains two components: i) an encoder that maps the input data to low-dimensional feature space and ii) a decoder that maps the low-dimensional representation to the reconstruction space. Both the encoder and the decoder are composed of multiple linear and nonlinear functions. Let $\mathbf{h}_{n,0}^{(v)} = \mathbf{x}_n^{(v)} \in \mathbb{R}^{d^{(v)}}$ be the n-th input data of the v-th view. Then, the output of l-th layer is denoted by

$$\mathbf{h}_{n,l}^{(v)} = f(W_{(l)}\mathbf{h}_{n,l-1}^{(v)} + \mathbf{b}_{(l)}) \in \mathbb{R}^{d_{(l)}}, \tag{6}$$

where f is the nonlinear activation function, $W_{(l)}$, $\mathbf{b}_{(l)}$, and $d_{(l)}$ denote the weight matrix, the bias vector of the encoder, and the number of neurons of the l-th layer, respectively. Moreover, for the input data of the v-th view $X^{(v)} = \{\mathbf{x}_1^{(v)}, \mathbf{x}_2^{(v)}, \ldots, \mathbf{x}_N^{(v)}\} \in \mathbb{R}^{d^{(v)} \times N}$, the corresponding reconstructions are

$$\hat{X}^{(v)} = \{\mathbf{h}_{1,L}^{(v)}, \mathbf{h}_{2,L}^{(v)}, \ldots, \mathbf{h}_{N,L}^{(v)}\}.$$

Then the problem is to minimize the data reconstruction loss

$$\min_{W_{(l)}, \mathbf{b}_{(l)}} \sum_{v=1}^{V} \{\frac{1}{2}\|X^{(v)} - \hat{X}^{(v)}\|_F^2 + \frac{1}{2}\sum_{l=1}^{L}(\|W_{(l)}\|_F^2 + \|\mathbf{b}_{(l)}\|_2^2)\}. \tag{7}$$

The first term is to minimize the reconstruction error between data points of each view. The second regularization term is to avoid overfitting. Though (7), we can explore the complex nonlinear structure of input data and get the low-dimensional latent representation by

$$H_{L/2}^{(v)} = \{\mathbf{h}_{1,L/2}^{(v)}, \mathbf{h}_{2,L/2}^{(v)}, \ldots, \mathbf{h}_{N,L/2}^{(v)}\}, \tag{8}$$

which can maintain the local detail information of each view.

3.2 Intra-view Global Similarity Structure Learning

The low-rank representation method can seek the similarity between samples by the self-expression learning. In fact, the latent representation $H_{L/2}^{(v)}$ is a compact representation of input samples. Thus, $H_{L/2}^{(v)}$ has the same clustering structure as input samples and also can explore the correlation between samples. Therefore, we construct the similarity matrix of each view on the low-dimensional latent representation. Moreover, we introduce the low-rank constraint as the *structure prior* to reflect the global similarity of samples in each view. The problem (7) is reformulated as

$$\min_{W_{(l)}, \mathbf{b}_{(l)}} \sum_{v=1}^{V} \{\frac{1}{2}\|X^{(v)} - \hat{X}^{(v)}\|_F^2 + \frac{1}{2}\sum_{l=1}^{L}(\|W_{(l)}\|_F^2 + \|\mathbf{b}_{(l)}\|_2^2)$$
$$+ \frac{\lambda_1}{2}\|H_{L/2}^{(v)} - H_{L/2}^{(v)}C_{L/2}^{(v)}\|_F^2 + \frac{\lambda_3}{2}\text{rank}(C_{L/2}^{(v)})\}, \tag{9}$$

where λ_1 and λ_3 are two positive tradeoff parameters and $C_{L/2}^{(v)}$ is the representation coefficient matrix. Here, $\|H_{L/2}^{(v)} - H_{L/2}^{(v)}C_{L/2}^{(v)}\|_F^2$ together with the low-rank term $\text{rank}(C_{L/2}^{(v)})$ preserve the global structure of each view.

Since the autoencoder consists of multi-layer nonlinear models, model (9) can explore the local nonlinear correlation between the input $X^{(v)}$ and the latent representation $H_{L/2}^{(v)}$. Therefore, a compact low-dimensional subspace representation of each view can be learned by the local and global similar structures of each view.

3.3 Inter-view Structure-Preserved Consistency Learning

By imposing the self-expression learning on the latent representation of each view, we get the corresponding representation matrix that expresses the global similarity between samples of each view. The samples from different views (i.e., the dataset $X^{(v)}$) yield different representation matrices (i.e., $C_{L/2}^{(v)}$). To cluster multi-view data, we need to explore the consistency among views, i.e., the correlation of different representation matrices.

Stacking the representation coefficients of all views into a third-order tensor, we explore the high-order correlation of views to obtain the consistency information among multiple views. Thus, we apply TNN on the stacked tensor

$$\min_{C, S} \|C\|_* + \sum_{v=1}^{V} \{\frac{\lambda_2}{2}\|C_{L/2}^{(v)} - S\|_F^2\} \tag{10}$$
$$\text{s.t. } C = \Phi(C_{L/2}^{(1)}, C_{L/2}^{(2)}, \ldots, C_{L/2}^{(v)}),$$

where Φ denotes the operator that stacks each representation matrix as the lateral slice of a third-order tensor, $\|C\|_*$ is TNN (see Def. 6) to characterize the high-order correlation of views, S is a unified affinity matrix learned form each matrix $C_{L/2}^{(v)}$, and λ_2 is a tradeoff parameter. By (10), we can simultaneously preserve the structure of each coefficient matrix $C_{L/2}^{(v)}$ and the consistency information among views.

3.4 Proposed Model

We integrate the above-mentioned hierarchical representation learning models into a unified framework and construct the following optimization problem

$$\min_{C, S, W_{(l)}, \mathbf{b}_{(l)}} \sum_{v=1}^{V} \{\frac{1}{2}\|X^{(v)} - \hat{X}^{(v)}\|_F^2 + \frac{\lambda_1}{2}\|H_{L/2}^{(v)} - H_{L/2}^{(v)}C_{L/2}^{(v)}\|_F^2$$
$$+ \frac{1}{2}\sum_{l=1}^{L}(\|W_{(l)}\|_F^2 + \|\mathbf{b}_{(l)}\|_2^2) + \frac{\lambda_2}{2}\|C_{L/2}^{(v)} - S\|_F^2\} + \frac{\lambda_3}{2}\|C\|_*$$
$$\text{s.t. } C = \Phi(C_{L/2}^{(1)}, C_{L/2}^{(2)}, \ldots, C_{L/2}^{(v)}). \tag{11}$$

By optimizing (11), we can obtain the affinity matrix, and then apply the spectral clustering tool [36] to get the multi-view clustering result. The advantages of our model are
• The proposed model can explore the complex nonlinear structure of each sample and consider both local and global structure priors for view-specific information, as well as the high-order correlation and consistency among all views.
• The learned low-dimensional latent subspace representation, the self-expression matrix, and the affinity matrix can boost each other in an interplay manner to get a satisfactory clustering performance.
• The proposed framework is flexible. The deep autoencoder module can be flexibly replaced by other deep neural network frameworks. Moreover, the high-order correlation of tensor can be characterized by other tensor low-rank constraints.

Next, we prove that the minimizer C has the tensor block diagonal structure, which implies a clear clustering structure.

Theorem 1. *(Tensor Block-diagonal Property). Assume that tensor subspaces $\{\mathcal{K}_1, \ldots, \mathcal{K}_k\}$ are independent, then the minimizer C of problem (11) is block-diagonal.*

Proof. Let C be the optimal solution of problem (11). We decompose $C = \Phi(C_{L/2}^{(1)}, C_{L/2}^{(2)}, \ldots, C_{L/2}^{(V)})$ into two parts, i.e., $C = \mathcal{B} + Q$, where $\mathcal{B}$ is a block diagonal tensor defined as

$$\mathcal{B}(i, j, :) = \begin{cases} C(i, j, :) & \textit{if the i-th and j-th samples are} \\ & \textit{drawn from the same subspace;} \\ 0 & \textit{otherwise.} \end{cases}$$

Assume that $\mathbf{x}_i^{(v)}$ belongs to the k-th tensor subspace $\mathcal{K}_k$, i.e., $\mathbf{x}_i^{(v)} \in \mathcal{K}_k$ and $\mathbf{h}_i^{(v)} \in \mathcal{K}_k$. According to the definitions of $\mathcal{B}$ and Q, we have $H^{(v)}\mathbf{b}_i^{(v)} \in \mathcal{K}_k$, and $H^{(v)}\mathbf{q}_i^{(v)} \in \bigoplus_{i \neq k} \mathcal{K}_i$, where $\mathbf{b}_i^{(v)}$ and $\mathbf{q}_i^{(v)}$ are the i-th columns of $B^{(v)}$ and $Q^{(v)}$, respectively. On the other hand, we have $H^{(v)}\mathbf{q}_i^{(v)} = H^{(v)}\mathbf{c}_i^{(v)} - H^{(v)}\mathbf{b}_i^{(v)} = \mathbf{h}_i^{(v)} - H^{(v)}\mathbf{b}_i^{(v)} \in \mathcal{K}_k$. Since subspaces $\mathcal{K}_i, i = 1, 2, \ldots, K$ are independent, $\mathcal{K}_k \bigcap \bigoplus_{k \neq i} \mathcal{K}_i = 0$. Thus, $H^{(v)}\mathbf{q}_i^{(v)} = 0$. We then have $H^{(v)}\mathbf{b}_i^{(v)} = \mathbf{h}_i^{(v)} - H^{(v)}\mathbf{q}_i^{(v)} = \mathbf{h}_i^{(v)}$ and $H^{(v)}B^{(v)} = H^{(v)}$. Hence, $\mathcal{B}$ is also the solution to (11).

In addition, since $C = \mathcal{B} + Q$, $\|C\|_* = \|\mathcal{B} + Q\|_* \geq \|\mathcal{B}\|_* + \|Q\|_*$, we obtain $\|C\|_* \geq \|\mathcal{B}\|_*$. Since C is the optimal solution of problem (11), then $\|C\|_* \leq \|\mathcal{B}\|_*$. Thus, we have $\|C\|_* = \|\mathcal{B}\|_*$ and the unique optimal solution $C = \mathcal{B}$ has the block-diagonal property. The proof is completed. □

Theorem 1 demonstrates the block-diagonal structure of C, which implies the underlying clustering structure, and the number of blocks is the number of clusters.

4 THE OPTIMIZATION ALGORITHM

We present an efficient alternative minimization algorithm based on the alternating direction method of multipliers framework (ADMM) [1, 44, 53] to solve problem (11).

By introducing the auxiliary variable $\mathcal{W} = C$, problem (11) can be reformulated as

$$\min_{C, \mathcal{W}, S, W_{(l)}, \mathbf{b}_{(l)}} \sum_{v=1}^{V} \{ \frac{1}{2}\|X^{(v)} - \hat{X}^{(v)}\|_F^2 + \frac{\lambda_1}{2}\|H_{L/2}^{(v)} - H_{L/2}^{(v)}C_{L/2}^{(v)}\|_F^2$$

$$+ \frac{1}{2}\sum_{l=1}^{L}(\|W_{(l)}\|_F^2 + \|\mathbf{b}_{(l)}\|_2^2) + \frac{\lambda_2}{2}\|C_{L/2}^{(v)} - S\|_F^2 \} + \frac{\lambda_3}{2}\|\mathcal{W}\|_*$$

$$\text{s.t. } C = \Phi(C_{L/2}^{(1)}, C_{L/2}^{(2)}, \ldots, C_{L/2}^{(v)}), \ \mathcal{W} = C. \quad (12)$$

Then, the augmented Lagrangian function of (12) is

$$L(C, \mathcal{W}, S, W_{(l)}, \mathbf{b}_{(l)}, \mathcal{P})$$

$$= \sum_{v=1}^{V} \{ \frac{1}{2}\|X^{(v)} - \hat{X}^{(v)}\|_F^2 + \frac{\lambda_1}{2}\|H_{L/2}^{(v)} - H_{L/2}^{(v)}C^{(v)}\|_F^2$$

$$+ \frac{1}{2}\sum_{l=1}^{L}(\|W_{(l)}\|_F^2 + \|\mathbf{b}_{(l)}\|_2^2) + \frac{\lambda_2}{2}\|C_{L/2}^{(v)} - S\|_F^2 \} \quad (13)$$

$$+ \frac{\lambda_3}{2}\|\mathcal{W}\|_* + \frac{\beta}{2}\|\mathcal{W} - C + \frac{\mathcal{P}}{\beta}\|_F^2,$$

where $\mathcal{P}$ is the Lagrangian variable and β is a penalty parameter. By ADMM algorithm, the original intractable optimization problem is transformed into several subproblems that are easy to solve. Next, we solve each of the following subproblems separately.

1. $(W_{(l)}, \mathbf{b}_{(l)})$-**subproblem.** We use stochastic gradient descent algorithm (SGD) [7, 35] to optimize the parameters involved in the deep autoencoder.

$$\arg \min_{(W_{(l)}, \mathbf{b}_{(l)})} \frac{1}{2}\|X^{(v)} - \hat{X}^{(v)}\|_F^2 + \frac{\lambda_1}{2}\|H_{L/2}^{(v)} - H_{L/2}^{(v)}C_{L/2}^{(v)}\|_F^2$$

$$+ \frac{1}{2}\sum_{l=1}^{L}(\|W_{(l)}\|_F^2 + \|\mathbf{b}_{(l)}\|_2^2). \quad (14)$$

The problem can be rewritten as the sample-wise form

$$\arg \min_{(W_{(l)}, \mathbf{b}_{(l)})} \frac{1}{2}\sum_{n=1}^{N}\|\mathbf{x}_n^{(v)} - \hat{\mathbf{x}}_n^{(v)}\|_2^2 + \frac{\lambda_1}{2}\|\mathbf{h}_{n, L/2}^{(v)} - H_{L/2}^{(v)}\mathbf{c}_{n, L/2}^{(v)}\|_2^2$$

$$+ \frac{1}{2}\sum_{l=1}^{L}(\|W_{(l)}\|_F^2 + \|\mathbf{b}_{(l)}\|_2^2), \quad (15)$$

where $\mathbf{x}_n^{(v)}$, $\mathbf{h}_{n, L/2}^{(v)}$, and $\mathbf{c}_{n, L/2}^{(v)}$ denote the n-th column vectors of $X^{(v)}$, $H_{L/2}^{(v)}$, and $C_{L/2}^{(v)}$, respectively.

By computing the sub-gradients of $W_{(l)}$ and $\mathbf{b}_{(l)}$, $\{W_{(l)}, \mathbf{b}_{(l)}\}_{l=1}^{L}$ can be updated by

$$W_{(l)} = W_{(l)} - \mu \frac{\partial f}{\partial W_{(l)}} \quad \text{and} \quad \mathbf{b}_{(l)} = \mathbf{b}_{(l)} - \mu \frac{\partial f}{\partial \mathbf{b}_{(l)}}, \quad (16)$$

where f is the objective function (15), $\frac{\partial f}{\partial W_{(l)}}$ and $\frac{\partial f}{\partial \mathbf{b}_{(l)}}$ are the sub-gradients of $W_{(l)}$ and $\mathbf{b}_{(l)}$, respectively, and μ is the learning rate.

2. $C_{L/2}^{(v)}$-**subproblem.** Optimizing $C_{L/2}^{(v)}$ is a quadratic problem

$$C_{L/2}^{(v)} = \arg \min_{C_{L/2}^{(v)}} \frac{\lambda_1}{2}\|H_{L/2}^{(v)} - H_{L/2}^{(v)}C_{L/2}^{(v)}\|_F^2 + \frac{\lambda_2}{2}\|C_{L/2}^{(v)} - S\|_F^2$$

$$+ \frac{\beta}{2}\|W^{(v)} - C_{L/2}^{(v)} + \frac{P^{(v)}}{\beta}\|_F^2, \quad (17)$$

where $W^{(v)} = \mathcal{W}(:, v, :)$, and $P^{(v)} = \mathcal{P}(:, v, :)$. The optimal solution satisfies the following equation

$$(\lambda_1 A^{(v)} + (\lambda_2 + \beta)I)C_{L/2}^{(v)} = \lambda_2 S + \lambda_1 A^{(v)} + \beta W^{(v)} + P^{(v)}, \quad (18)$$

where $A^{(v)} = (H^{(v)})^\top H^{(v)}$.

3. S-**subproblem.** Fixed other variables, the S-subproblem is

$$S = \arg \min_{S} \sum_{v=1}^{V} \frac{\lambda_2}{2}\|C_{L/2}^{(v)} - S\|_F^2, \quad (19)$$

where S can be obtained by

$$S = \sum_{v=1}^{V} C_{L/2}^{(v)}/V. \quad (20)$$

4. $\mathcal{W}$-**subproblem.** Fixed other variables, the $\mathcal{W}$-subproblem can be formulated as

$$\mathcal{W} = \arg \min_{\mathcal{W}} \frac{\lambda_3}{2}\|\mathcal{W}\|_* + \frac{\beta}{2}\|\mathcal{W} - C + \frac{\mathcal{P}}{\beta}\|_F^2. \quad (21)$$

Algorithm 1 The optimization algorithm for solving (11).

Input: The input multi-view data $\{X^{(v)}\}_{v=1}^{V}$, cluster number c, the embedding dimension d of the latent representation, and parameters $\lambda_1, \lambda_2, \lambda_3, \beta = 10^{-4}$, and $\beta_{max} = 10^{10}$.

Initialization: $C, \mathcal{W}, \mathcal{P}, S$ and neural network $\{W^{(l)}, \mathbf{b}^{(l)}\}_{l=1}^{L}$.

1: **while** not converged and $t \leq 100$ **do**
2:     Update $W^{(l)}$ and $\mathbf{b}^{(l)}$ via (16);
3:     Update $C^{(v)}$ via (18);
4:     Update S via (20);
5:     Update $\mathcal{W}$ by (22);
6:     Update $\mathcal{P}$ by (23);
7:     Update penalty parameter β by $\beta = \min(1.1 \times \beta, \beta_{max})$;
8:     Check the convergence condition:
      $\|\mathcal{W} - C\|_\infty \leq 10^{-7}$
9: **end while**
10: Performing the spectral clustering tool on the unified affinity matrix S.

Output: The clustering results.

It can be solve by tensor singular value thresholding operator [51]

$$\mathcal{W} = \mathcal{U}\Gamma_{\frac{\lambda_3}{\beta}}(\mathcal{S})\mathcal{V}^T, \tag{22}$$

where $(\mathcal{U}, \mathcal{S}, \mathcal{V})$ comes from the TSVD of $(C - \frac{\mathcal{P}}{\beta})$ and $\Gamma_{\frac{\lambda_3}{\beta}}(\mathcal{S}) =$ diag($\max\{S - \frac{\lambda_3}{\beta}, 0\}$).

5. $\mathcal{P}$**-subproblem** The variable $\mathcal{P}$ can be updated by

$$\mathcal{P} = \mathcal{P} + \beta(\mathcal{W} - C). \tag{23}$$

Finally, we summarize the proposed algorithm in Algorithm 1.

Complexity Analysis. The main computational cost depends on the training cost of the neural network and the computational cost of C and $\mathcal{W}$-subproblems. For the parametric network, the computational complexity is $O(aN^3)$, where a denotes the number of training epochs. Since the C-subproblem involves the inverse operation of a matrix ($N \times N$), the computational complexity is $O(N^3)$. The $\mathcal{W}$-subproblem contains the TSVD of a tensor with size $N \times V \times N$, where both the complexities of FFT and inverse FFT are $O(2NVN \log N)$, and the SVD computation of a matrix ($N \times V$) costs $O(NV^2)$. Thus, the complexity of $\mathcal{W}$-subproblem is $O(2N^2V \log N + N^2V^2)$. Let t denote the number of iterations. Thus, the total computational complexity of Algorithm 1 is

$$O(t(aN^3 + 2N^2V \log N + N^2V^2)).$$

Convergence Analysis. The general ADMM algorithm solves the optimal problem with two block variables [1, 8]. Since the proposed model contains multiple block variables, it is nontrivial to prove its theoretical convergence. Fortunately, without loss of generality, the calculation of $W^{(l)}$ and $\mathbf{b}^{(l)}$ in Algorithm 1 is convergent. Moreover, other variables in Algorithm 1 have closed-form solutions. Therefore, the proposed algorithm is empirically convergent according to the analysis in previous studies [42, 43]. We also demonstrate the numerical convergence of the proposed algorithm in Section 6.

5 NUMERICAL EXPERIMENTS

In this section, we evaluate the proposed HRMC by extensive experiments on both simulated and real-world multi-view datasets.

5.1 Experiment Settings

5.1.1 Baselines. We compare the proposed method with eight state-of-the-art baselines: TSVD based Multi-view Subspace Clustering (**tSVD-MSC**) [42] enforces the low-rank tensor constraint on the rotated tensor to obtain the consensus among multi-view data; Reciprocal Multi-layer Subspace Learning (**RMSL**) [22] reduces the dimension of original data by hierarchical self-representative layers and uses the backward encoding network to enforce the consensus among data samples; Weighted Tensor-Nuclear Norm Minimization (**WTNNM**) [12] introduces the weighted TNN to explore the high-order correlation between views; Multi-view Clustering in Latent Embedding Space (**MCLES**) [3] learns the global consistency structure on the latent embedding representation; Multi-view Dimensionality co-Reduction (**MDcR**) [47] uses the kernel function to obtain the low-dimensional feature of each view and employees the Hilbert-Schmidt independence criterion to capture the correlation between views; Projective Low-Rank Subspace Clustering (**PLrSC**) [21] learns the low-rank representation of all data by a deep encoder, which is trained on a small part of the large original dataset; Deep Matrix Factorization for MVC (**DMF-MVC**) [52] introduces the deep semi-nonnegative matrix factorization to learn the hidden feature and uses the graph regularizer to learn the share representation; Learnable Subspace Clustering (**LeaSC**) [20] achieves the low-dimensional representation of data according to a trained parametric function, where the function is established by a robust predictive coding machine.

5.1.2 Evaluation Measures. To evaluate the proposed method, we use six metrics to measure the clustering performance, including accuracy (ACC), normalized mutual information (NMI), adjusted rand index (AR), F-score, Precision, and Recall [30]. The higher values of them indicate the better clustering performance. For avoiding the randomness, we run 10 independent trials on datasets to obtain the final clustering results in terms of means and variances. We test all algorithms on the platform with an Intel(R) Core(TM) i7-8700M CPU with 3.70 GHz and 32 GB of RAM.

5.1.3 Parameter Setting. There are three regularization parameters in our model, i.e., λ_1, λ_2, and λ_3. The values of λ_1 and λ_3 are selected from $\{0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1\}$, and $\lambda_2 = 1$. In addition, we fix the learning rate $\mu = 2 \times 10^{-10}$. For baselines, we carefully adjust parameters to obtain the excellent clustering performance according to the suggestions of relevant articles.

5.1.4 Implementation Details. In the proposed algorithm, we choose tanh as the nonlinear activation function of deep autoencoder. For real-world dataset, we construct a five-layer neural network that consists of $d^{(v)} - 200 - 100 - 200 - d^{(v)}$ neurons, where $d^{(v)}$ is the feature dimension of the input data. Since the deep autoencoder involves multiple nonlinear hidden layers, it is difficult to optimize its weights. According to [14], for the initialization of the network, we adopt two steps of pre-training and fine-tuning. Using a layer-by-layer learning strategy, only one layer of features is pre-trained at a time, and then the deep network is initialized with pre-trained

Table 1: Clustering performance on simulated dataset. The best and second best values are in bold and <u>underlined</u>, respectively.

Method	ACC	NMI	AR	F-score	Precision	Recall
tSVD-MSC	0.9150±0.0000	0.8515±0.0000	0.8121±0.0000	0.8599±0.0000	0.8427±0.0000	0.8778±0.0000
RMSL	0.4050±0.0028	0.1530±0.0036	0.0804±0.0025	0.3518±0.0011	0.3002±0.0012	0.4497±0.0189
WTNNM	<u>0.9963±0.0000</u>	<u>0.9842±0.0000</u>	<u>0.9900±0.0000</u>	<u>0.9925±0.0000</u>	<u>0.9925±0.0000</u>	<u>0.9926±0.0000</u>
MCLES	0.7080±0.0000	0.6305±0.0000	0.5495±0.0000	0.6685±0.0000	0.6306± 0.0000	0.7112±0.0000
MDcR	0.9200±0.0000	0.7988±0.0000	0.8085±0.0000	0.8568±0.0000	0.8471±0.0000	0.8667±0.0000
PLrSC	0.7851±0.0086	0.6803±0.0072	0.6177±0.0146	0.7188±0.0075	0.6855±0.0109	0.7578±0.0048
DMF-MVC	0.8925±0.0000	0.7662±0.0000	0.7483±0.0000	0.8122±0.0000	0.7972±0.0000	0.8276±0.0000
LeaSC	0.9504±0.0001	0.8546±0.0004	0.8754±0.0005	0.9064±0.0003	0.9060± 0.0003	0.9069±0.0003
HRMC	**1.0000±0.0000**	**1.0000±0.0000**	**1.0000±0.0000**	**1.0000±0.0000**	**1.0000±0.0000**	**1.0000±0.0000**

weights. For the initialization of the structure prior $C^{(v)}$, the sparse subspace clustering method [10] is adopted in our algorithm.

5.2 Simulated Data Experiments

We first conduct experiments on simulated datasets to evaluate the effectiveness of the proposed method.

5.2.1 Simulated Data Generation. The details of generating simulated data are as follows: 1) we generate four types of datasets, each containing 200 samples in 2D space, as the low-dimensional latent representations; 2) we use different nonlinear activation functions on four datasets to generate the original multi-view dataset. The mathematical form is

$$x_n^{(v)} = (\tanh(\phi^{(v)}(Ah_n)))^2,$$

where $h_n \in \mathbb{R}^2$ is the low-dimensional data, $A \in \mathbb{R}^{100 \times 2}$ is a random matrix obeying the Gaussian distribution, and $x_n^{(v)} \in \mathbb{R}^{100}$ is the high-dimensional data with complex structure. Set $\{\phi^{(v)}\}_{v=1}^{V}$ denotes different nonlinear functions for different views. In the following, we let $\phi^{(1)}$ = sigmoid, $\phi^{(2)}$ = tanh, and $\phi^{(3)}$ = ReLU and generate the simulated multi-view data with three views.

5.2.2 Results. Table 1 presents the clustering performance of different methods on the simulated multi-view data. We observe that the proposed HRMC achieves the best clustering result compared with other baselines. By considering the weights of singular values on the original feature space, WTNNM obtained the second best clustering result. LeaSC also achieved the third best performance due to the trained parametric function. The performance of t-SVD-MSC, RMSL, and MCLES is unsatisfactory. One possible reason is that they only consider the consensus information of different views while ignoring the view-specific local nonlinear structure. Moreover, the performance of MDcR is suboptimal due to ignorance of the high-order correlations between views.

Fig. 2 shows the visualizations of different methods. We find that the proposed HRMC obtains very close samples of the same class and far-apart samples of different classes. Moreover, Fig. 3 illustrates the block diagonal structures of unified similarity matrices by different methods. Compared with other baselines, our HRMC obtains the clearer block diagonal structure, which is consistent with the above results and also confirms Theorem 1.

5.3 Real-world Data Experiments

We then evaluate the effectiveness of the proposed HRMC on four common multi-view datasets described by multiple features.
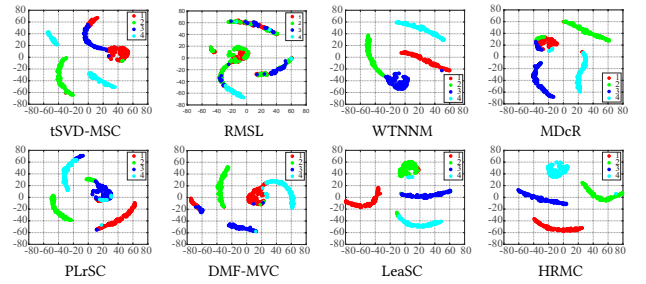
Figure 2: Visualizations of different clustering methods on simulated dataset.
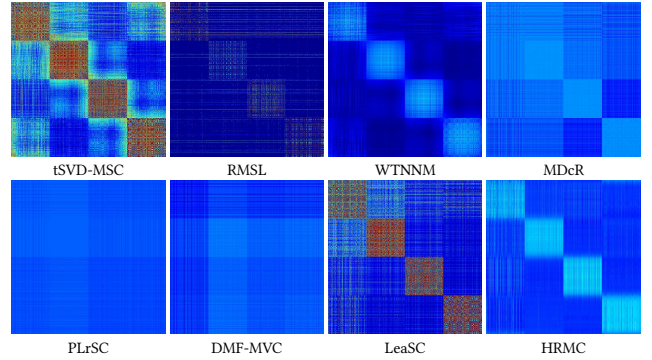
Figure 3: Block diagonal structures of the similarity matrices of different clustering methods on simulated dataset.

5.3.1 Datasets. **100Leaves**[1] contains $1,600$ images of 100 classes. Three features are extracted, including shape descriptor, fine-scale margin, and texture histogram.

MSRC[2] has 210 objects with seven classes, including cars, trees, airplanes, buildings, bicycles, cows, and human faces. Each of them is described by three views.

BBCnews[2] is composed of 685 reports, containing business, politics, sports, technology, and entertainment. Each of them has four views.

Webkb[2] consists of 203 web-pages of four classes. Each web-page is described by the content of the page, the anchor text of the hyper-link, and the text of the title.

5.3.2 Results. Table 2 lists the clustering performance by all algorithms. We observe that the proposed HRMC achieves the best performance among all the methods in terms of ACC, NMI, AR, F-score, Precision, and Recall on all the cases. The detailed analysis is given as follows:

(1) Methods tSVD-MSC, MCLES, and WTNNM only consider the consistency between views and the global structure of each view while ignoring the local nonlinear structure of each sample. Although DMF-MVC and LeaSC can learn rich hierarchical information, they overemphasize the local information and reconstruction information of the input data, thereby ignoring the global structure of each view and high-order correlation of multi-view data. By contrast, the proposed HRMC fully explores the consistency between

---

[1]https://cs.nyu.edu/roweis/data.html
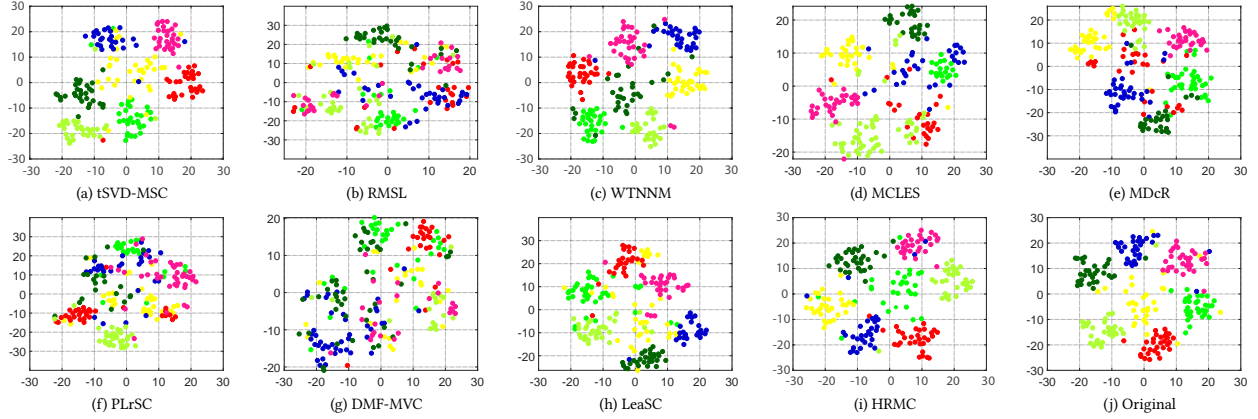[2]https://github.com/wangsiwei2010/awesome-multi-view-clustering

Figure 4: Visualizations of different clustering methods on MSRC dataset.

Table 2: The clustering performance by all algorithms on the 100Leaves, BBCnews, MSRC, and Wekbk dataset.

Datasets	Method	ACC	NMI	AR	F-score	purity	recall
100Leaves	tSVD-MSC	0.8991±0.0001	0.9648±0.0000	0.8908±0.0001	0.8916±0.0001	0.8490±0.0002	0.9385±0.0000
	RMSL	0.6287±0.0003	0.7835±0.0001	0.4695±0.0005	0.4749±0.0004	0.4384±0.0005	0.5182±0.0004
	WTNNM	0.9078±0.0001	0.9638±0.0000	0.8939±0.0001	0.8947±0.0001	0.8584±0.0002	0.9341±0.0000
	MCLES	0.7819±0.0023	0.7622±0.0004	0.7402±0.0018	0.7098±0.0003	0.7071±0.0014	0.6798±0.0032
	MDcR	0.7244±0.0002	0.8705±0.0001	0.6323±0.0004	0.6359±0.0003	0.6042±0.0003	0.6712±0.0004
	PLrSC	0.7511±0.0007	0.8884±0.0001	0.6761±0.0009	0.6793±0.0008	0.6454±0.0012	0.7173±0.0005
	DMF-MVC	0.2546±0.0000	0.5606±0.0000	0.1065±0.0000	0.1156±0.0000	0.1069±0.0000	0.1258±0.0000
	LeaSC	0.7262±0.0003	0.8675±0.0001	0.6318±0.0004	0.6354±0.0004	0.6047±0.0004	0.6695±0.0004
	HRMC	**0.9144±0.0000**	**0.9783±0.0000**	**0.9046±0.0000**	**0.9055±0.0000**	**0.8680±0.0000**	**0.9465±0.0000**
BBCnews	tSVD-MSC	0.7169±0.0000	0.8709±0.0000	0.7423±0.0000	0.7986±0.0000	0.8727±0.0000	0.7360±0.0000
	RMSL	0.3322±0.0006	0.0318±0.0000	0.0056±0.0000	0.3410±0.0007	0.2410±0.0000	0.6074±0.0283
	WTNNM	0.7188±0.0000	0.8709±0.0000	0.7425±0.0000	0.7987±0.0000	0.8727±0.0000	0.7363±0.0000
	MCLES	0.7191±0.0171	0.6562±0.0177	0.5569±0.0458	0.6901±0.0185	0.5889±0.0283	0.8654±0.0029
	McDR	0.9007±0.0000	0.8351±0.0000	0.8595±0.0000	0.8934±0.0000	0.8832±0.0000	0.9039±0.0000
	PLrSC	0.3360±0.0024	0.0837±0.0017	0.0472±0.0007	0.2576±0.0004	0.2782±0.0004	0.2399±0.0005
	DMF-MVC	0.3220±0.0000	0.0573±0.0000	0.0317±0.0000	0.2690±0.0000	0.2618±0.0000	0.2766±0.0000
	LeaSC	0.3757±0.0000	0.0592±0.0000	0.0112±0.0000	0.3861±0.0000	0.2429±0.0000	**0.9409±0.0000**
	HRMC	**0.9688±0.0000**	**0.9064±0.0000**	**0.9231±0.0000**	**0.9412±0.0000**	**0.9505±0.0000**	0.9322±0.0000
MSRC	tSVD-MSC	0.9619±0.0000	0.9381±0.0000	0.9195±0.0000	0.9307±0.0000	0.9287±0.0000	0.9327±0.0000
	RMSL	0.5514±0.0029	0.4077±0.0021	0.2998±0.0019	0.3978±0.0014	0.3944±0.0014	0.4013±0.0014
	WTNNM	0.9810±0.0000	**0.9650±0.0000**	**0.9562±0.0000**	**0.9623±0.0000**	0.9607±0.0000	**0.9639±0.0000**
	MCLES	0.7133±0.0009	0.6828±0.0010	0.5508±0.0017	0.6181±0.0012	0.5733±0.0013	0.6708±0.0012
	MDcR	0.7790±0.0000	0.6853±0.0001	0.6013±0.0001	0.6569±0.0001	0.6537±0.0001	0.6601±0.0001
	PLrSC	0.5924±0.0037	0.4877±0.0025	0.3732±0.0032	0.4616±0.0024	0.4543±0.0024	0.4693±0.0024
	DMF-MVC	0.5093±0.0000	0.3543±0.0000	0.2188±0.0000	0.3329±0.0000	0.3164±0.0000	0.3512±0.0000
	LeaSC	0.7752±0.0000	0.6442±0.0000	0.5643±0.0000	0.6254±0.0000	0.6188±0.0000	0.6322±0.0000
	HRMC	**0.9810±0.0000**	0.9604±0.0000	0.9558±0.0000	0.9619±0.0000	**0.9613±0.0000**	0.9626±0.0000
Webkb†	tSVD-MSC	0.7143±0.0000	0.4132±0.0000	0.4503±0.0000	0.6650±0.0000	0.6695±0.0000	0.6605±0.0000
	RMSL	0.4655±0.0020	0.0373±0.0002	0.0151±0.0006	0.4936±0.0028	0.3922±0.0001	0.6891±0.0267
	WTNNM	0.7192±0.0000	0.4190±0.0000	0.4564±0.0000	0.6693±0.0000	0.6720±0.0000	0.6667±0.0000
	MCLES	-	-	-	-	-	-
	McDR	0.6695±0.0000	0.3289±0.0000	0.3746±0.0000	0.6026±0.0000	0.6499±0.0000	0.5617±0.0000
	PLrSC	0.4567±0.0005	0.2078±0.0055	0.1444±0.0039	0.4284±0.0007	0.5133±0.0030	0.3709±0.0012
	DMF-MVC	-	-	-	-	-	-
	LeaSC	0.6744±0.0000	0.2595±0.0000	0.2894±0.0000	0.5979±0.0000	0.5412±0.0000	0.6679±0.0000
	HRMC	**0.7635±0.0001**	**0.4160±0.0000**	**0.4841±0.0000**	**0.6926±0.0000**	**0.6738±0.0000**	**0.7125±0.0000**

† Since the Webkb data is too sparse to compute SVD, we do not list the results of MCLES and DMF-MVC.

views and the local nonlinear structure as well as global similarity structure of each view, thereby resulting in higher clustering evaluation performance.

(2) Although the kernel-based multi-view clustering methods like MDcR can learn the nonlinear subspace structure and obtain the low-dimensional feature of original data, MDcR has unsatisfactory clustering performance on 100Leaves, MSRC, and Webkb datasets. This is because the kernel function-based method is too rigorous in the choice of the kernel function, and an inappropriate kernel may lead to poor clustering results. However, the proposed HRMC uses the deep autoencoder to adaptively learn the nonlinear structure within input data and the latent representation, thereby performing better than kernel-based methods.

Fig. 4 shows the visualization of different methods on MSRC dataset, where different colors represent different clusters of samples. We observe that the sample distributions of Fig. 4(c) (WTNNM)

and Fig. 4(i) (our HRMC) have clearer boundaries than other baselines. Further comparing our HRMC with WTNNM as shown in Fig. 4(c), the distance between clusters obtained by our HRMC in Fig. 4(i) is larger. Moreover, the clustering results of the proposed HRMC in Fig. 4(i) are the closest to the original ones in Fig. 4(j). The visual comparison shows the superiority of the proposed method.

6 DISCUSSIONS

This section discusses the effects of different regularizations, the influence of parameters, and the numerical convergence of our HRMC.

6.1 Ablation Study

In the proposed method, the autoencoder is used to reconstruct the view-specific local information, the self-expression term is used to preserve the view-specific global structure, and the low-rank tensor constraint focuses on exploring the higher-order correlation between views. We evaluate the effect of each term by different variants of our HRMC: HRMC without (w/o) nonlinear term, HRMC w/o global structure, and HRMC w/o high-order correlation term.

Table 3 lists the clustering performance of different methods on the simulated multi-view data (in Section 5.2.1). We observe from Table 3 that the proposed HRMC achieves the best clustering performance though its variant w/o high-order correlation term has the second best performance and the other two variants have the inferior performance. These results indicate that both the view-specific local nonlinear structure and global structure priors play an important role in enhancing the clustering performance.

Fig. 5 shows the visualization of clustering performance. It is shown that the proposed HRMC can barely cluster the simulated multi-view data without the nonlinear term or the global structure. Provided with the low-rank tensor constraint (i.e., the high-order correlation), the performance is further improved. This suggests that these three terms complement and strengthen each other.

6.2 Parameter Analysis

The proposed method contains four parameters λ_1, λ_2, λ_3, and β. In this part, we investigate the influence of different parameters on the proposed method. Taking the Wekbk dataset as an example, Fig. 6

Table 3: Clustering performance of HRMC and its variants.

Method	ACC	NMI	AR	F-score	Precision	Recall
HRMC w/o nonlinear term	0.2813	0.0075	0.0026	0.2769	0.2508	0.3092
HRMC w/o global structure	0.2512	0.0074	0.0020	0.3982	0.2491	0.9925
HRMC w/o high-order correlation	0.9587	0.8895	0.8965	0.9224	0.9185	0.9264
HRMC	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**

(a) HRMC w/o nonlinear term

(b) HRMC w/o global structure

(c) HRMC w/o high-order correlation
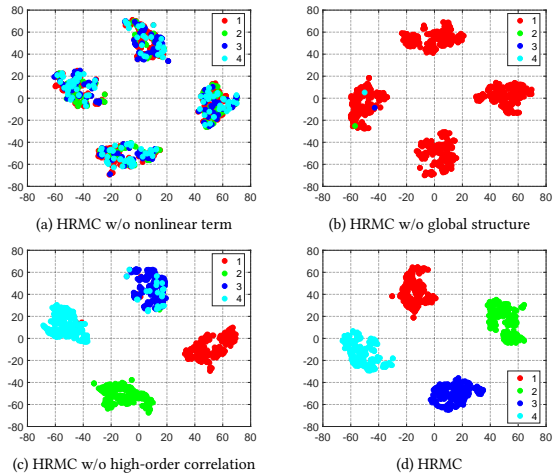
(d) HRMC

**Figure 5: Effect of each term on HRMC for data clustering.**

plots the ACC and NMI curves with respect to λ_1, λ_2, λ_3, and β. We observe that our HRMC is stable with varied parameters λ_2 and β. When λ_2 and β vary in [1, 10] and [0.0001, 2], respectively, the proposed HRMC can achieve the promising performance. Moreover, we observe that the proposed method is sensitive with parameters λ_1 and λ_3. Thus, we empirically set $\lambda_2 = 1$, $\beta = 0.0001$, and tune λ_1 and λ_3 in [0.001, 0.1] with interval 0.004.

6.3 Convergence Analysis

We take BBCnews and Webkb datasets as examples and plot the relative error curves as shown in Fig. 7. We observe that the relative error curves (vs. number of iterations) rapidly decrease to zero, suggesting that the proposed algorithm numerically achieves convergence.

7 CONCLUSION

In this work, we proposed a hierarchical representation learning method with integration of intra-sample adaptive latent representation learning, intra-view global similarity structure learning, and inter-view structure-preserved consistency learning for MVC. We introduced the deep autoencoder to capture the nonlinear structure in each sample, and learned the self-repression of latent representation to characterize the global similarity structure among samples of a view. Moreover, we applied the tensor low-rank constraint to explore the high-order correlation among views. The proposed method can fully capture both local and global structures of view-specific and high-order correlation among views to obtain the clustering results. Extensive numerical results demonstrate the advantages of the proposed method in MVC. As one of future studies, our algorithm will be further extended to large-scale datasets.
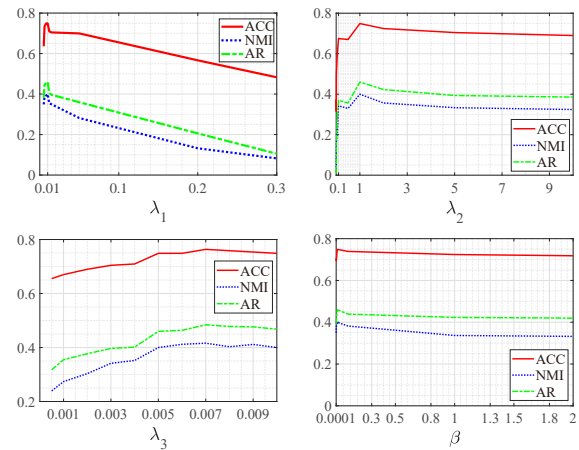
Figure 6: The influence of λ_1, λ_2, λ_3, and β on Wekbk dataset.

(a) BBCnews

(b) Webkb
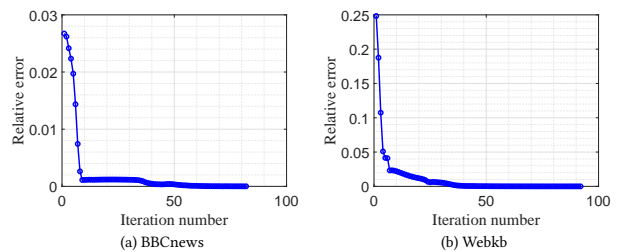
Figure 7: The relative error curves with respect to iterations.

ACKNOWLEDGMENTS

REFERENCES

[1] H. Attouch, J. Bolte, and B. F. Svaiter. 2011. Convergence of descent methods for semi-algebraic and tame poblems: Proximal algorithms, forward-backward splitting, and regularized Gauss-Seidel methods. *Math. Program.* 137, 1 (2011), 91–129. https://doi.org/article/10.1007/s10107-011-0484-9

[2] M. Brbic and I. Kopriva. 2018. Multi-view low-rank sparse subspace clustering. *Pattern Recognit.* 73 (2018), 247–258. https://doi.org/abs/1708.08732

[3] M. S. Chen, L. Huang, C. D. Wang, and D. Huang. 2020. Multi-view clustering in latent embedding space. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 3513–3520. https://doi.org/10.1609/aaai.v34i04.5756

[4] Y. Y. Chen, X. L. Xiao, and Y. C. Zhou. 2019. Multi-view clustering via simultaneously learning graph regulzarized low-rank tensor representation and affinity matrix. In *Proceedings of International Conference on Multimedia and Expo.* 1348–1353. https://doi.org/10.1109/ICME.2019.00234

[5] Y. Y. Chen, X. L. Xiao, and Y. C. Zhou. 2020. Jointly learning kernel representation tensor and affinity matrix for multi-view clustering. *IEEE Trans. Multimedia* 22, 8 (2020), 1985–337. https://doi.org/10.1109/TMM.2019.2952984

[6] M. Cheng, L. Jing, and M. K. Ng. 2019. Tensor-based low-dimensional representation learning for multi-view clustering. *IEEE Trans. Image Process.* 28, 5 (2019), 2399–2414. https://doi.org/10.1109/TIP.2018.2877937

[7] C. Darken, J. Chang, and J. Moody. 1992. Learning rate schedules for faster stochastic gradient search. In *Neural Networks for Signal Processing II Proceedings of the 1992 IEEE Workshop.* 1–11. https://doi.org/icsi/node/3218

[8] M. Ding, T-Z. Huang, X-L. Zhao, Michael. K. Ng, and T-H. Ma. 2021. Tensor train rank minimization with nonlocal self-similarity for tensor completion. *Inverse Probl. Imaging* 15, 3 (2021), 475–498. https://doi.org/10.3934/ipi.2021001

[9] G. W. Du, L. H. Zhou, Y. D. Yang, K. Lu, and L. Z. Wang. 2021. Deep multiple auto-encoder-based multi-view clustering. *Data Science and Engineering* 6 (2021), 323–337. https://doi.org/article/10.1007/s41019-021-00159-z

[10] E. Elhamifar and R. Vidal. 2013. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 11 (2013), 2765–2781. https://doi.org/10.1109/TPAMI.2013.57

[11] L. L. Fu, J. H. Yang, C. Chen, and C. F. Zhang. 2022. Low-rank tensor approximation with local structure for multi-view intrinsic subspace clustering. *Inf. Sci.* 606 (2022), 877–891.

[12] Q. X. Gao, W. Xia, Z. Z. Wan, D. Y. Xie, and P. Zhang. 2020. Tensor-SVD based graph learning for multi-view subspace clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 3930–3937. https://doi.org/10.1609/aaai.v34i04.5807

[13] B. Haeffele, C. You, and R. Vidal. 2021. A critique of self-expressive deep subspace clustering. In *Proceedings of International Conference on Learning Representations*. 1–20. https://doi.org/10.48550/arXiv.2010.03697

[14] G. E. Hinton and R. R. Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *Science* 313, 5786 (2006), 504–507. https://doi.org/126/science.1127647

[15] S. Huang, Z. Kang, and Z. Xu. 2020. Auto-weighted multi-view clustering via deep matrix decomposition. *Pattern Recogn.* 97 (2020), 107015. https://doi.org/10.1016/j.patcog.2019.107015

[16] P. Ji, T. Zhang, H. D. Li, M. Salzmann, and I. Reid. 2017. Deep subspace clustering networks. In *Proceedings of the International Conference on Neural Information Processing Systems*. 23–32. https://doi.org/doi/abs/10.5555/3294771.3294774

[17] M. Kheirandishfard, F. Zohrizadeh, and F. Kamangar. 2020. Multi-level representation learning for deep subspace clustering. In *Proceedings of Winter Conference on Applications of Computer Vision*. 2039–2048. https://doi.org/10.48550/arXiv.2001.08533

[18] B. Z. Li, X. L. Zhao, J. L. Wang, Y. Chen, T. X. Jiang, and J. Liu. 2021. Tensor Completion via Collaborative Sparse and Low-Rank Transform. *IEEE Trans. Comput. Imaging* 7 (2021), 1289–1303. https://doi.org/10.1109/TCI.2021.3126232

[19] C. S. Li, C. Yang, B. Liu, Y. Yuan, and G. R. Wang. 2021. LRSC: Learning Representations for Subspace Clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 8340–8348. https://doi.org/index.php/AAAI/article/view/17014

[20] J. Li, H. F. Liu, Z. Q. Tao, H. D. Zhao, and Y. Fu. 2022. Learnable subspace clustering. *IEEE Trans. Neural. Netw. Learn. Syst.* 33, 3 (2022), 1119–1133. https://doi.org/10.1109/TNNLS.2020.3040379

[21] J. Li, H. F. Liu, H. D. Zhao, and Y Fu. 2017. Projective low-rank subspace clustering via learning deep encoder. In *Proceedings of the International Joint Conference on Artificial Intelligence*. 2145–2151. https://doi.org/10.24963/ijcai.2017/298

[22] R. H. Li, C. Q. Zhang, H. Z. Fu, X. Peng, T. Y. Zhou, and Q. H. Hu. 2019. Reciprocal multi-layer subspace learning for multi-view clustering. In *Proceedings of the International Conference on Computer Vision*. 8172–8180. https://doi.org/10.1109/ICCV.2019.00826

[23] Z. L. Li, C. Tang, X. W. Liu, X. Zheng, G. H. Yue, W. Zhang, and E. Zhu. 2021. Consensus Graph Learning for Multi-view Clustering. *IEEE Trans. Image Process.* 24 (2021), 2461–2472. https://doi.org/10.1109/TMM.2021.3081930

[24] B. Q. Lin, Y. Xie, Y. Y. Qu, C. H. Li, and X. D. Liang. 2018. Jointly deep multi-view learning for clustering analysis. In *Proceedings of International Conference on Computer Vision and Pattern Recognition*. 1–9. https://doi.org/10.48550/arXiv.1808.06220

[25] G. Liu, Q. Liu, and P. Li. 2017. Blessing of dimensionality: Recovering mixture data via dictionary pursuit. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 1 (2017), 47–60. https://doi.org/10.1109/TPAMI.2016.2539946

[26] G. C. Liu, Z. C. Lin, S. C. Yan, J. Sun, Y. Yu, and Y. Ma. 2013. Robust recovery of subspace structures by low-rank representation. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 1 (2013), 171–184. https://doi.org/10.1109/TPAMI.2012.88

[27] C. Lu, H. Min, Z. Zhao, L. Zhu, D. Huang, and S. Yan. 2012. Robust and efficient subspace segmentation via least squares regression. In *Proceedings of the European Conference on Computer Vision*. 347–360. https://doi.org/10.48550/arXiv.1404.6736

[28] Y. Luo, D. Tao, C. Ramamohanarao, C. Xu, and Y. Wen. 2015. Tensor canonical correlation analysis for multi-view dimension reduction. *IEEE Trans. Knowl. Data En.* 27 (2015), 3111–3124. https://doi.org/10.1109/TKDE.2015.2445757

[29] J. C. Lv, Z. Kang, X. Lu, and Z. L. Xu. 2021. Pseudo-supervised deep subspace clustering. *IEEE Trans. Image Process.* 30 (2021), 5252–5263. https://doi.org/10.1109/TIP.2021.3079800

[30] C. D. Manning, P. Raghavan, and H. Schutze. 2008. Introduction to information retrieval. *Cambridge University Press, Cambridge* 1 (2008). https://doi.org/IR-book/information-retrieval-book.html

[31] J. Masci, U. Meier, D. Ciresan, and J. Schmidhuber. 2011. Stacked convolutional auto-encoders for hierarchical feature extraction. In *Proceedings of International Conference on Artificial Neural Networks*. 52–59. https://doi.org/10.1007/978-3-642-21735-7_7

[32] F. Nie, G. Cai, and X. Li. 2017. Multi-view clustering and semisupervised classification with adaptive neighbours. In *Proceedings of the AAAI Conference on Artificial*

[33] *Intelligence.* 2408–2414. https://doi.org/index.php/AAAI/article/view/10909

[33] X. Peng, J. Feng, S. Xiao, W. Y. Yau, J. T. Zhou, and S. Yang. 2018. Structured autoencoders for subspace clustering. *IEEE Trans. Image Process.* 27, 10 (2018), 5076–5086. https://doi.org/10.1109/TIP.2018.2848470

[34] X. Peng, Z. Y. Huang, J. C. Lv, H. Y. Zhu, and T. Y. Zhou. 2019. COMIC: Multi-view ClusteringWithout Parameter Selection. In *Proceedings of the International Conference on Machine Learning*. 5092–5101. https://doi.org/v97/peng19a.html

[35] N. Qian. 1999. On the momentum term in gradient descent learning algorithms. *Neural networks: the official journal of the International Neural Network Society* 12, 1 (1999), 145–151. https://doi.org/10.1016/S0893-6080(98)00116-6

[36] J. Shi and J. Malik. 2000. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 22, 8 (2000), 888–905. https://doi.org/10.1109/34.868688

[37] K. W. Tang, R. S. Liu, Z. X. Su, and J. Zhang. 2014. Structure-constrained low-rank representation. *IEEE Trans. Neural Netw. Learn. Syst.* 25, 12 (2014), 2167–2179. https://doi.org/10.1109/TNNLS.2014.2306063

[38] W. Wang, R. Arora, K. Livescu, and J. Bilmes. 2015. On deep multiview representation learning. In *Proceedings of the International Conference on Machine Learning*. 1083–1092. https://doi.org/doi/10.5555/3045118.3045234

[39] J. Wu, Z. Lin, and H. Zha. 2019. Essential tensor learning for multi-view spectral clustering. *IEEE Trans. Image Process.* 28, 12 (2019), 5910–5922. https://doi.org/10.1109/TIP.2019.2916740

[40] X. L. Xiao, Y. Y. Chen, Y. J. Gong, and Y. C. Zhou. 2021. Prior knowledge regularized multiview self-representation and its applications. *IEEE Trans. Neural. Netw. Learn. Syst.* 32, 3 (2021), 1325–1338. https://doi.org/10.1109/TNNLS.2020.2984625

[41] J. Y. Xie, R. Girshick, and A. Farhadi. 2016. Unsupervised deep embedding for clustering analysis. In *Proceedings of International Conference on Machine Learning*. 478–487. https://doi.org/10.5555/3045390.3045442

[42] Y. Xie, D. C. Tao, W. S. Zhang, Y. Liu, L. Zhang, and Y. Y. Qu. 2018. On unifying multi-View self-representations for clustering by tensor multi-rank minimization. *International Journal of Computer Vision* 126, 11 (2018), 1157–1179. https://doi.org/10.1007/s11263-018-1086-2

[43] Y. Xie, W. S. Zhang, Y. Y. Qu, L. Q. Dai, and D. C. Tao. 2020. Hyper-Laplacian regularized multilinear multi-view self-representations for clustering and semi-supervised learning. *IEEE Trans. Cyber.* 50, 2 (2020), 572–586. https://doi.org/10.1109/TCYB.2018.2869739

[44] J. H. Yang, X. L. Zhao, T. Y. Ji, T. H. Ma, and T. Z. Huang. 2020. Low-rank tensor train for tensor robust principal component analysis. *Appl. Math. Comput.* 367, 15 (2020), 124783. https://doi.org/10.1016/j.amc.2019.124783

[45] X. Yang, C. Deng, F. Zheng, J. C. Yan, and W. Liu. 2019. Deep spectral clustering using dual autoencoder network. In *Proceedings of Conference on Computer Vision and Pattern Recognition*. 15–20. https://doi.org/10.1109/CVPR.2019.00419

[46] C. Q. Zhang, H. Z. Fu, Q. H. Hu, X. C. Cao, Y. Xie, D. C. Tao, and D. Xu. 2020. Generalized latent multi-view subspace clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* 42, 1 (2020), 86–99. https://doi.org/10.1109/TPAMI.2018.2877660

[47] C. Q. Zhang, H. Z. Fu, Q. H. Hu, P. F. Zhu, and X. C. Cao. 2017. Flexible multi-view dimensionality co-reduction. *IEEE Trans. Image Process.* 26, 2 (2017), 648–659. https://doi.org/10.1109/TIP.2016.2627806

[48] C. Q. Zhang, Q. H. Hu, H. Z. Fu, P. F. Zhu, and X. C. Cao. 2017. Latent multi-view subspace clustering. In *Proceedings of Conference on Computer Vision and Pattern Recognition*. 4279–4287. https://doi.org/10.1109/CVPR.2017.461

[49] D. J. Zhang, Y. F. Sun, B. Eriksson, and L. Balzano. 2017. Deep unsupervised clustering using mixture of autoencoders. (2017). https://doi.org/abs/1712.07788

[50] J. Zhang, X. Li, P. Jing, J. Liu, and Y. Su. 2018. Low-rank regularized heterogeneous tensor decomposition for subspace clustering. *IEEE Signal Process. Lette.* 25, 3 (2018), 333–337. https://doi.org/10.1109/LSP.2017.2748604

[51] X. Q. Zhang, Z. Y. Zhou, D. Wang, and Y. Ma. 2014. Hybrid singular value thresholding for tensor completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 1362–1368. https://doi.org/index.php/AAAI/article/view/8898

[52] H. D. Zhao, Z. M. Ding, and Y. Fu. 2017. Multi-view clustering via deep matrix factorization. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 2921–2927. https://doi.org/ocs/index.php/AAAI/AAAI17/paper/view/14647

[53] X. L. Zhao, J. H. Yang, T. H. Ma, T. X. Jiang, M. K. Ng, and T. Z. Huang. 2022. Tensor Completion via Complementary Global, Local, and Nonlocal Priors. *IEEE Trans. Image Process.* 31 (2022), 984–999. https://doi.org/10.1109/TIP.2021.3138325

[54] L. L. Zhen, D. Z. Peng, W. Wang, and X. Yao. 2020. Kernel truncated regression representation for robust subspace clustering. *Inform. Sciences* 524 (2020), 59–76. https://doi.org/10.1016/j.ins.2020.03.033

[55] L. Zhou, X. Bai, D. Wang, X. L. Liu, J. Zhou, and E. Hancock. 2019. Deep subspace clustering via latent distribution preserving. In *Proceedings of International Joint Conference on Artificial Intelligence*. 4440–4446. https://doi.org/10.24963/ijcai.2019/617

[56] P. Zhou, C. Y. Lu, J. S. Feng, Z. C. Lin, and S. C. Yan. 2021. Tensor low-rank representation for data recovery and clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* 43, 5 (2021), 1718–1732. https://doi.org/10.1109/TPAMI.2019.2954874