This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2022.3230964

Low-Rank Tensor Learning for Incomplete Multiview Clustering

Jie Chen, *Member, IEEE*, Zhu Wang, Hua Mao, and Xi Peng, *Member, IEEE*

Abstract—Incomplete multiview clustering (IMVC) is an effective way to identify the underlying structure of incomplete multiview data. Most existing algorithms based on matrix factorization, graph learning or subspace learning have at least one of the following limitations: (1) the global and local structures of high-dimensional data are not effectively explored simultaneously; (2) the high-order correlations among multiple views are ignored. In this paper, we propose a low-rank tensor learning (LRTL) method that learns a consensus low-dimensional embedding matrix for IMVC. We first take advantage of the self-expressiveness property of high-dimensional data to construct sparse similarity matrices for individual views under low-rank and sparsity constraints. Individual low-dimensional embedding matrices can be obtained from the sparse similarity matrices using spectral embedding techniques. This approach simultaneously explores the global and local structures of incomplete multiview data. Then, we present a multiview embedding matrix fusion model that incorporates individual low-dimensional embedding matrices into a third-norm tensor to achieve a consensus low-dimensional embedding matrix. The fusion model exploits complementary information by finding the high-order correlations among multiple views. In addition, the computational cost of an improved fusion strategy is dramatically reduced. Extensive experimental results demonstrate that the proposed LRTL method outperforms several state-of-the-art approaches.

Index Terms—Multiview clustering, tensor nuclear norm, spectral embedding, high-dimensional data.

1 Introduction

In real scenarios, high-dimensional data are often collected from different signal sources or represented by different types of features [1]. For example, a color image can be described by different modalities, e.g., its color, textures, and edges. A piece of news can be reported in several languages while still delivering the same message. These examples are referred to as a typical kind of multiview data. Multiview clustering (MVC) attempts to partition samples into their respective groups by fully integrating the information obtained from multiple views. In contrast to clustering with a single view, clustering with multiple views may provide some consistency and complementary information regarding multiview data, which can effectively improve clustering performance [2], [3], [4], [5], [6].

Most MVC algorithms always assume that the data of all views are fully available and that the instances are strictly aligned in these views [7], [8], [9], [10], [11], [12], [13]. For example, Chen *et al.* presented a symmetric multiview low-rank representation method to seek the low-dimensional structures of high-dimensional data across multiple views [10]. Zhu *et al.* presented a multiview spectral clustering method to generate a common affinity matrix for MVC [12]. Xie *et al.* [14] presented a deep multi-view joint clustering framework to simultaneously learn multiple deep embedded features, a multiview fusion mechanism, and clustering assignments for deep MVC. In practical applications, multiview data often suffer from the absence of instances in some views for various reasons, e.g., temporary failures on data acquisition devices or high data collection costs. How to efficiently manipulate incomplete multiview data becomes the incomplete MVC (IMVC) problem. The lack of instances in some views inevitably degrades the clustering performance of traditional MVC algorithms.

A variety of IMVC algorithms have been proposed during the past decade to alleviate the problem of missing instances in multiple views [15], [16], [17]. Most IMVC algorithms are roughly grouped into five categories from the perspective of the theory behind their optimization models, i.e., subspace learning-based methods [18], [19], [20], [21], nonnegative matrix factorization (NMF)-based methods [22], [23], graph learning-based methods [15], [17], multiple kernel-based methods [16], [24] and deep learning-based methods [25], [26], [27], [28], [29]. A number of IMVC algorithms often construct a shared similarity matrix for spectral clustering or learn a consensus clustering matrix for the k-means algorithm [30] across multiple views by inferring the missing views or imputing the missing features or incomplete base kernel matrices. For example, Wen *et al.* [20] presented an incomplete multiview tensor-based spectral clustering method that incorporates feature space-based missing view inference and similarity graph learning into a unified framework to learn similarity matrices for multiple views. Liu *et al.* [16] presented an IMVC algorithm that learns a consensus clustering matrix by imputing incomplete kernel matrices, which are initially generated by

- *J. Chen and X. Peng are with the College of Computer Science, Sichuan University, Chengdu 610065, China (E-mail: chenjie2010@scu.edu.cn; pengx.gm@gmail.com).*
- *Z. Wang is with the Law School, Sichuan University, Chengdu 610065, China. E-mail: wangzhu@scu.edu.cn.*
- *H. Mao is with the Department of Computer and Information Sciences, Northumbria University, Newcastle, NE1 8ST, U. K. (E-mail: hua.mao@northumbria.ac.uk).*

Manuscript received April 23, 2022; revised September 6, 2022 and November 16, 2022. This work was supported in part by National Natural Science Foundation of China (NSFC) under Grant 61303015, in part by National Key Project under Grant GJXM92579, and in part by National Key Research and Development Program of China under Grant 2018YFC0830300. (Corresponding author: Zhu Wang.)

incomplete views. These algorithms can obtain encouraging clustering results for incomplete multiview data. However, some limitations and drawbacks remain. First, the missing instances in the sample views are not effectively recovered when individual instances are completely absent in each individual view. As the missing rate gradually increases, the numbers of instances available in multiple views also decline considerably. As a result, conducting imputation on the original missing instances or filling the missing instances with zeros or mean values could adversely affect the clustering performance of such methods, especially for a large missing data ratio. Next, the IMVC algorithms that take advantage of the self-expressiveness property of high-dimensional data cannot fully capture the global and local structures of such data because they only consider the intuitive combinations of various matrix norms, e.g., low-rank or sparsity norms, in high-dimensional data. However, it is not simple to explicitly determine which norms play dominant roles in joint data self-representations. Finally, the high-order correlations among multiple views are often ignored in IMVC. Consequently, IMVC still faces significant challenges.

Lu *et al.* [31] recently introduced a tensor-robust principal component analysis (PCA) method to recover a low tubal rank tensor and sparse tensor from their sum, which is based on the tensor-tensor product [32]. Motivated by advances in tensor analysis techniques [31], [32], several works have introduced tensor nuclear norms to exploit the high-order correlations among multiple views [9], [17], [21], [33]. Most of these methods stack the LRRs of multiple views into an individual tensor to exploit the high-order information embedded in these views [17], [21], [33], and they apply tensor-singular value decomposition (t-SVD) or its variants on the self-expressive tensor to obtain individual LRRs for multiple views. For example, Xie *et al.* [33] proposed a hyper-Laplacian regularized multilinear multiview self-representation method to learn the correlations among multiple views in a unified tensor space. The multiview self-representation approach is considered an excellent LRR extension for multiple views. Li *et al.* [17] proposed an IMVC method that stacks low-rank dimensional embedding matrices into a third-order tensor and rotates the tensor to learn a consensus clustering representation. This IMVC method performs imputation on the missing parts of the spectral embedding matrices under a low-rank tensor constraint. However, the reasons why the rotation of the third-order tensor is beneficial for finding high-order correlations deserve further investigation.

In this paper, we present a low-rank tensor learning (LRTL) method for IMVC. Different from with most existing IMVC methods, the proposed LRTL approach simultaneously explores the global and local structures of incomplete multiview data, which is beneficial for capturing consistency information across multiple views. Specifically, we first take advantage of the self-expressiveness property of high-dimensional data to learn sparse similarity matrices for individual views under low-rank and sparsity constraints. In particular, the missing instances are removed from the views when learning the sparse similarity matrices. A global block diagonal structure is investigated for sparse similarity matrices. Then, we apply spectral embedding techniques on the sparse similarity matrices to obtain individual low-dimensional embedding matrices. To find the positive high-order correlations of multiple views, we present a multiview embedding matrix fusion model by incorporating individual low-dimensional embedding matrices into a third tensor. This is beneficial for capturing complementary information among the instances of multiple views. Finally, the proposed multiview embedding matrix fusion model achieves a consensus low-dimensional embedding matrix for k-means clustering. In addition, we present an alternative fusion strategy to reduce the computational cost of the optimization problem in the fusion model. Simultaneously, a theoretical analysis is given to explain why the fusion model can work effectively under certain conditions.

TABLE 1
Definitions of symbols.

Symbols	Definitions
$\mathbf{I}_k$	The identity matrix of size $k \times k$
$\mathbf{X}^T$	The transpose of $\mathbf{X}$
$\mathbf{X}^{-1}$	The inverse of $\mathbf{X}$
$diag(\mathbf{X})$	The vector containing the n diagonal elements of $\mathbf{X}$
$tr(\mathbf{X})$	The trace of $\mathbf{X}$
$\|\mathbf{X}\|_0$	The number of nonzero elements in $\mathbf{X}$
$\|\mathbf{X}\|_1$	The l_1-norm of $\mathbf{X}$
$\|\mathbf{X}\|_F$	The Frobenius norm of $\mathbf{X}$
$\|\mathbf{X}\|_*$	The nuclear norm of $\mathbf{X}$

Our major contributions are summarized as follows:

- The proposed approach learns individual low-dimensional embedding matrices from incomplete multiview data by considering low-rank and sparsity constraints. This technique simultaneously explores the global and local structures of multiview data.
- The proposed approach presents a multiview embedding matrix fusion model, which exploits complementary information by finding the high-order correlations of multiple views, to achieve a consensus low-dimensional embedding matrix.
- Our method presents an alternative fusion strategy for the fusion model. This strategy explains why the fusion model is able to work effectively under certain conditions and simultaneously achieve a reduced computational cost.
- Experimental results on benchmark datasets indicate that the proposed method achieves considerable improvements over the state-of-the-art IMVC approaches.

The remainder of this paper is organized as follows. In Section 2, we provide a brief review of the related work. Section 3 presents the proposed LRTL method in detail. Extensive experiments are conducted to validate the effectiveness of the proposed LRTL method in Section 4. Finally, we conclude this paper in Section 5.

2 Related Work

In this section, we briefly review the current spectral embedding techniques and tensor nuclear norm theory. For convenience, the definitions of the utilized symbols are shown in Table 1.

2.1 Spectral Embedding Techniques

Consider a matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n] \in \mathbb{R}^{d\times n}$ with n samples, where d is the dimensionality of each sample. The weighted adjacency matrix is $\mathbf{W} \in \mathbb{R}^{n\times n}$, where W_{ij} represents a nonnegative weight that measures the similarity between samples $\mathbf{x}_i$ and $\mathbf{x}_j$. A normalized Laplacian matrix $\mathbf{L} \in \mathbb{R}^{n\times n}$ is defined as follows:

$$\mathbf{L} = \mathbf{I}_n - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2} \tag{1}$$

where $\mathbf{D} = diag\left[d_1, d_2, ..., d_n\right]$ is a diagonal matrix with elements $d_i = \sum\limits_{j=1}^{n} W_{ij}$ [34].

Assume that the samples can be partitioned into k distinct clusters. A low-dimensional embedding matrix $\mathbf{H} \in \mathbb{R}^{n\times k}$ can be obtained by solving the following optimization problem of the normalized cut (NCut) [34]:

$$\min_{\mathbf{H}} tr\left(\mathbf{H}^T\mathbf{L}\mathbf{H}\right) \quad s.t. \quad \mathbf{H}^T\mathbf{H} = \mathbf{I}_k. \tag{2}$$

The solution $\mathbf{H}$ consists of the eigenvectors of the normalized Laplacian matrix $\mathbf{L}$ that correspond to the k smallest eigenvalues.

Each row vector of $\mathbf{H}$ is normalized by the l_2-norm, i.e., $\overline{\mathbf{H}} = \mathbf{P}\mathbf{H}$, where $\mathbf{p}_i$ is the i-th row of $\mathbf{H}$ and $\mathbf{P} = diag\left[p_1, p_2, ..., p_n\right]$ is a diagonal matrix with elements $p_i = \frac{1}{\sqrt{\mathbf{h}_i^T\mathbf{h}_i}}$. Ideally, $W_{ij} = 0$ if samples $\mathbf{x}_i$ and $\mathbf{x}_j$ are in different clusters. Then,

$$\overline{\mathbf{H}} = \mathbf{Y}\mathbf{R} \tag{3}$$

where each column of $\mathbf{Y} \in \mathbb{R}^{n\times k}$ is an indicator vector [34] and $\mathbf{R} \in \mathbb{R}^{k\times k}$ is an orthogonal matrix. Thus,

$$\overline{\mathbf{H}}\,\overline{\mathbf{H}}^T = \mathbf{Y}\mathbf{R}(\mathbf{Y}\mathbf{R})^T = \mathbf{Y}\mathbf{Y}^T. \tag{4}$$

Here, $\overline{\mathbf{H}}\,\overline{\mathbf{H}}^T$ is a block diagonal matrix [17], i.e.,

$$\overline{\mathbf{H}}\,\overline{\mathbf{H}}^T = \begin{bmatrix} \overrightarrow{\mathbf{1}_{n_1}} & 0 & \cdots & 0 \\ 0 & \overrightarrow{\mathbf{1}_{n_2}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \cdots \\ 0 & 0 & \cdots & \overrightarrow{\mathbf{1}_{n_k}} \end{bmatrix} \tag{5}$$

where n_i $(1 \leq i \leq k)$ represents the number of samples in the i-th cluster and $\overrightarrow{\mathbf{1}_{n_i}}$ a submatrix of size $n_i \times n_i$ containing all ones.

2.2 Tensor Nuclear Norm Theory

Given a tensor $\mathcal{Y} \in \mathbb{R}^{n_1\times n_2\times n_3}$, we define

$$unfold\left(\mathcal{Y}\right) = \begin{bmatrix} \mathcal{Y}^{(1)} \\ \mathcal{Y}^{(2)} \\ \vdots \\ \mathcal{Y}^{(n_3)} \end{bmatrix}, \quad fold\left(unfold\left(\mathcal{Y}\right)\right) = \mathcal{Y} \tag{6}$$

where the $unfold$ operator maps $\mathcal{Y}$ to a matrix of size $n_1n_2 \times n_3$ and $fold$ is its inverse operator [35]. The block circulant matrix $bcirc\left(\mathcal{Y}\right)$ is defined as

$$bcirc\left(\mathcal{Y}\right) = \begin{bmatrix} \mathcal{Y}^{(1)} & \mathcal{Y}^{(n_3)} & \cdots & \mathcal{Y}^{(2)} \\ \mathcal{Y}^{(2)} & \mathcal{Y}^{(1)} & \cdots & \mathcal{Y}^{(3)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{Y}^{(n_3)} & \mathcal{Y}^{(n_3-1)} & \cdots & \mathcal{Y}^{(1)} \end{bmatrix}. \tag{7}$$

Algorithm 1 The t-SVT operator [31]

Input: $\mathcal{Y} \in \mathbb{R}^{n_1\times n_2\times n_3}$ and a parameter $\alpha > 0$.
Compute $\bar{\mathcal{Y}} = fft\left(\mathcal{Y}, [], 3\right)$;
for $i = 1, ..., \frac{\lceil n_3+1 \rceil}{2}$ **do**
    $[\mathbf{U}, \mathbf{S}, \mathbf{V}] = svd\left(\bar{\mathbf{Y}}^{(i)}\right)$;
    $\bar{\mathbf{W}}^{(i)} = \mathbf{U}\left(\mathbf{S} - \alpha\right)_+ \mathbf{V}^T$;
end for
for $i = \frac{\lceil n_3+1 \rceil}{2} + 1, ..., n_3$ **do**
    $\bar{\mathbf{W}}^{(i)} = conj\left(\bar{\mathbf{W}}^{(n_3-i+2)}\right)$;
end for
Output: $\mathcal{D}_\alpha\left(\mathcal{Y}\right) = ifft\left(\bar{\mathcal{W}}, [], 3\right)$.

The t-product of two tensors $\mathcal{A} \in \mathbb{R}^{n_1\times n_2\times n_3}$ and $\mathcal{B} \in \mathbb{R}^{n_2\times n_1\times n_3}$ is defined as

$$\mathcal{E} = \mathcal{A} * \mathcal{B} = fold\left(bcirc\left(\mathcal{A}\right) \cdot unfold\left(\mathcal{B}\right)\right) \tag{8}$$

where $\mathcal{E} \in \mathbb{R}^{n_1\times n_l\times n_3}$ [35]. The t-product is equivalent to matrix multiplication in the Fourier domain [35]. The t-SVD operation is defined as

$$\mathcal{Y} = \mathcal{U} * \mathcal{S} * \mathcal{V} \tag{9}$$

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 $\mathcal{S} \in \mathbb{R}^{n_1\times n_2\times n_3}$ is an f-diagonal tensor whose frontal slices are diagonal matrices [31], [35].

The tensor nuclear norm of $\mathcal{Y}$ is defined as

$$\left\|\mathcal{Y}\right\|_* = \sum_{i=1}^{r} \mathcal{S}\left(i, i, 1\right) \tag{10}$$

where r represents the tensor tubal rank of $\mathcal{Y}$. The tensor tubal rank is equivalent to the number of nonzero singular values in $\mathcal{Y}$ [31]. The problem of finding a low tubal rank approximation of $\mathcal{Y}$ can be formulated as

$$\min_{\mathcal{X}\in\mathbb{R}^{n_1\times n_2\times n_3}} \alpha\left\|\mathcal{X}\right\|_* + \frac{1}{2}\left\|\mathcal{X} - \mathcal{Y}\right\|_F^2. \tag{11}$$

This problem has a closed-form solution, which is the proximal operator of the matrix nuclear norm. The tensor-singular value thresholding (t-SVT) operator is defined as follows:

$$\mathcal{D}_\alpha\left(\mathcal{Y}\right) = \mathcal{U} * \mathcal{S}_\alpha * \mathcal{V} \tag{12}$$

where $\mathcal{S}_\alpha = ifft\left(\left(\bar{\mathcal{S}} - \alpha\right)_+, [], 3\right)$, $\bar{\mathcal{S}}$ is a real tensor, $ifft$ is a MATLAB command and $(t)_+ = max(t, 0)$ [31]. The t-SVT operator is a proximity operator that is closely related to the tensor nuclear norm. Specifically, this operator applies a shrinkage-based thresholding rule to the singular values $\bar{\mathcal{S}}$ of the frontal slices of $\mathcal{Y}$. For completeness, the details of the t-SVT operator are given in Algorithm 1 [31], where fft is also a MATLAB command.

3 Low-Rank Tensor Learning for IMVC

In this section, we present an LRTL method that learns a consensus low-dimensional embedding matrix for IMVC. The proposed LRTL method is mainly composed of two parts: sparse LRR learning and multiview embedding matrix fusion. Moreover, we present an alternative fusion strategy to improve the computational efficiency of the LRTL method.

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2022.3230964

3.1 Problem Formulation

Given a set of incomplete multiview data $\left\{\mathbf{X}^{(v)} \in \mathbb{R}^{d_v \times n}, v \in \{1,2,...n_v\}\right\}$ with n samples, $\mathbf{X}^{(v)}$ is the v-th view of the incomplete multiview data. Each view $\mathbf{X}^{(v)}$ has n instances, i.e., $\mathbf{X}^{(v)} = \left[\mathbf{x}_1^{(v)}, \mathbf{x}_2^{(v)}, ..., \mathbf{x}_n^{(v)}\right]$, and d_v represents the instance dimensionality in the v-th view. We use a diagonal indicator matrix $\mathbf{M}^{(v)}$ to denote the missing instances in the v-th view, which is defined as:

$$M_{ii}^{(v)} = \begin{cases} 1, \text{ the instance } \mathbf{x}_i^{(v)} \text{ is available in the } v\text{-th view} \\ 0, \text{ otherwise.} \end{cases} \tag{13}$$

The goal of IMVC is to group the n samples into k clusters according to particular similarity measures.

3.2 Sparse LRR Learning for Spectral Embedding

For ease of exploration, we begin with an assumption that each sample in $\mathbf{X}$ is exactly drawn from k independent subspaces, i.e., $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2..., \mathbf{X}_k]$. We consider the following general nuclear norm minimization problem:

$$\min_{\mathbf{Z}} \|\mathbf{Z}\|_* \quad s.t. \quad \mathbf{X} = \mathbf{XZ} \tag{14}$$

where $\mathbf{Z}$ is considered the LRR of $\mathbf{X}$ with respect to itself. The optimal solution to Problem (14) is

$$\mathbf{Z}^* = \mathbf{X}^\dagger \mathbf{X} \tag{15}$$

where $\mathbf{X}^\dagger$ is the pseudoinverse of $\mathbf{X}$. In particular, $\mathbf{Z}^*$ is a block diagonal, i.e.,

$$\mathbf{Z}^* = \begin{bmatrix} \mathbf{Z}_1^* & 0 & 0 & 0 \\ 0 & \mathbf{Z}_2^* & 0 & 0 \\ & & \ddots & \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \mathbf{Z}_k^* \end{bmatrix} \tag{16}$$

where $rank\left(\mathbf{Z}_i^*\right) = rank\left(\mathbf{X}_i\right) \; (1 \le i \le k)$.

Suppose that $\mathbf{Z}^*$ is regarded as a weighted adjacency matrix for $\mathbf{X}$, where Z_{ij} represents the weight of the similarity between samples $\mathbf{x}_i$ and $\mathbf{x}_j$. Thus, the normalized Laplacian matrix $\mathbf{L}$ constructed from $\mathbf{Z}^*$ is also a block diagonal matrix. Moreover, $\mathbf{HH}^T$ is block diagonal according to solving Problem (2). However, $\mathbf{X}$ may not strictly follow independent subspace structures in practice. This results in the block diagonal property of $\mathbf{HH}^T$ not being satisfied. Consequently, it is crucially important to find an adjacency matrix for the incomplete multiview data $\left\{\mathbf{X}^{(v)}\right\}_{v=1}^{n_v}$ that is as approximate to the block diagonal structure as possible.

Recovering the missing instances in the given views is an intractable problem. Moreover, it is impossible to construct complete LRRs for incomplete multiview data $\left\{\mathbf{X}^{(v)}\right\}_{v=1}^{n_v}$ due to missing instances. Hence, the existing instances are preserved, and the columns of $\mathbf{X}^{(v)}$ corresponding to the missing instances are removed from the v-th view. Suppose $\mathbf{X}_c^{(v)} \in \mathbb{R}^{d_v \times N_v}$ consists of the existing instances in the vth view, where N_v is the number of columns of $\mathbf{X}_c^{(v)}$. To seek the corresponding adjacency matrix, we explore the global structure of the incomplete multiview data by using LRR. Specifically, we consider the nuclear norm minimization problem for $\left\{\mathbf{X}_c^{(v)}\right\}_{v=1}^{n_v}$:

$$\min_{\mathbf{Z}_c^{(v)}} \sum_{v=1}^{n_v} \left\|\mathbf{Z}_c^{(v)}\right\|_* + \lambda \left\|\mathbf{X}_c^{(v)} - \mathbf{X}_c^{(v)}\mathbf{Z}_c^{(v)}\right\|_F^2 \tag{17}$$

where $\mathbf{Z}_c^{(v)} \in \mathbb{R}^{N_v \times N_v}$ is an LRR for the existing instances in the vth view. The variable $\mathbf{Z}_c^{(v)}$ associated with the vth view can be updated independently in Problem (17). Let $\mathbf{X}_c^{(v)} = \mathbf{U}_c^{(v)}\mathbf{\Sigma}_c^{(v)}\mathbf{V}_c^{(v)T}$ be the SVD of $\mathbf{X}_c^{(v)}$, where $diag\left(\mathbf{\Sigma}_c^{(v)}\right) = [r_1, r_2, ...r_n]$ and $r_i \; (1 \le i \le n)$ represents the corresponding singular value in $\mathbf{\Sigma}_c^{(v)}$. Each subproblem with respect to $\mathbf{Z}_c^{(v)}$ has a closed-form solution [36], i.e.,

$$\widetilde{\mathbf{Z}}_{\mathbf{c}}^{(\mathbf{v})} = \mathbf{V}_1^{(v)}\left(\mathbf{I} - \frac{1}{\lambda}\left(\mathbf{\Sigma}_1^{(v)}\right)^{-2}\right)\mathbf{V}_1^{(v)T} \tag{18}$$

where $\mathbf{V}_c^{(v)} = \left[\mathbf{V}_1^{(v)}, \mathbf{V}_2^{(v)}\right]$, $\mathbf{V}_1^{(v)}$ consists of the left c columns of $\mathbf{V}_c^{(v)}$ according to the sets $\left\{c : r_c > \frac{1}{\sqrt{\lambda}}\right\}$, $\mathbf{V}_2^{(v)}$ is composed of the remaining columns of $\mathbf{V}_c^{(v)}$, and $diag\left(\mathbf{\Sigma}_1^{(v)}\right) = [r_1, r_2, ...r_c]$.

According to the above discussion, each LRR in $\left\{\widetilde{\mathbf{Z}}_{\mathbf{c}}^{(\mathbf{v})}\right\}_{v=1}^{n_v}$ may not be block diagonal but of low rank instead. We construct $\mathbf{Z}^{(v)} = \mathbf{U}\mathbf{\Sigma}^{\frac{1}{2}}$, where the economy SVD of $\widetilde{\mathbf{Z}}_{\mathbf{c}}^{(\mathbf{v})}$ is $\widetilde{\mathbf{Z}}_{\mathbf{c}}^{(\mathbf{v})} = \mathbf{U\Sigma V}^T$. By taking advantage of the angular information contained in the columns of $\mathbf{Z}^{(v)}$ $\left(\mathbf{Z}^{(v)} = \left[\mathbf{z}_1^{(v)}, \mathbf{z}_2^{(v)}, ..., \mathbf{z}_{N_v}^{(v)}\right]\right)$, we define a weighted adjacency matrix $\mathbf{C}_c^{(v)} \in \mathbb{R}^{N_v \times N_v}$ to evaluate the similarity among the existing instances in the vth view. Each element in $\mathbf{C}_c^{(v)}$ is defined as follows [37]:

$$\left[\mathbf{C}_c^{(v)}\right]_{ij} = \left(\frac{\mathbf{z}_i^{(v)}\left(\mathbf{z}_j^{(v)}\right)^T}{\left\|\mathbf{z}_i^{(v)}\right\|_2\left\|\mathbf{z}_j^{(v)}\right\|_2}\right)^2 \tag{19}$$

where $\mathbf{z}_i^{(v)}$ and $\mathbf{z}_j^{(v)}$ represent the ith and jth columns of $\mathbf{Z}^{(v)}$, respectively. The introduction of the angular information from the columns of $\mathbf{Z}^{(v)}$ can help to alleviate the negative effects brought by deviations from the ideal entries in $\mathbf{Z}^{(v)}$.

The size of $\mathbf{C}_c^{(v)}$ may be different from other matrices in multiple views due to the differences in the numbers of missing instances among multiple views. Thus, $\mathbf{C}^{(v)} \in \mathbb{R}^{n \times n}$ is employed to represent the complete weighted adjacency matrix that measures the similarity among all the instances in the vth view. We introduce a mapping function for each weighted adjacency matrix of the incomplete multiple views, which is defined as

$$\mathbf{C}^{(v)} = h\left(\mathbf{C}_c^{(v)}, \mathbf{M}^{(v)}\right) \tag{20}$$

where the mapping function $h(\cdot)$ performs an operation in which all elements of $\mathbf{C}_c^{(v)}$ are filled into the corresponding entries of $\mathbf{C}^{(v)}$ and the other entries of $\mathbf{C}^{(v)}$ are filled with zeros.

Algorithm 2 Calculating individual low-dimensional embedding matrices $\left\{\mathbf{H}^{(v)}\right\}_{v=1}^{n_v}$

Input: Data matrices $\left\{\mathbf{X}_c^{(v)}\right\}_{v=1}^{n_v}$ and a parameter $\lambda > 0$.
1: **for** $v = 1$ to n_v **do**
2: Calculate $\widetilde{\mathbf{Z}}_{\mathbf{c}}^{(\mathbf{v})}$ by solving Problem (17) using Equation (18);
3: Calculate $\left[\mathbf{C}_c^{(v)}\right]_{ij}$ for each pair of existing instances by using Equation (19);
4: Calculate $\mathbf{C}^{(v)}$ by using Equation (20);
5: $\mathbf{C}^{(v)} \leftarrow normalize_{(0,1]}\left(\mathbf{C}^{(v)}\right)$;
6: Calculate $\mathbf{W}^{(v)}$ by solving Problem (22) using the Euclidean projection method with a scalar $z = 1$ [38];
7: $\mathbf{W}^{(v)} \leftarrow \left(\mathbf{W}^{(v)} + \mathbf{W}^{(v)T}\right)\Big/2$;
8: Calculate $\mathbf{L}^{(v)}$ by using Equation (23);
9: Calculate $\mathbf{H}^{(v)}$ by solving Problem (2).
10: **end for**
Output:
$\left\{\mathbf{H}^{(v)}\right\}_{v=1}^{n_v}$.

We further explore the local structure of the incomplete multiview data by pursuing the sparsity of $\mathbf{C}^{(v)}$. This optimization problem is formulated as

$$\begin{aligned} &\min_{\mathbf{W}^{(v)}} \left\|\mathbf{W}^{(v)}\right\|_1 + \eta \left\|\mathbf{W}^{(v)} - \mathbf{C}^{(v)}\right\|_F^2 \\ &s.t. \quad W_{ij}^{(v)} \geq 0,\ \left(\mathbf{W}_i^{(v)}\right)^T \mathbf{1} = 1 \end{aligned} \tag{21}$$

where η is a tradeoff parameter, $\mathbf{W}_i^{(v)}$ is the ith column of $\mathbf{W}^{(v)} \in \mathbb{R}^{n\times n}$ and $W_{ij}^{(v)}$ is the jth element of $\mathbf{W}_i^{(v)}$. The sparsity regularization term $\left\|\mathbf{W}^{(v)}\right\|_1$ in Problem (21) becomes a constant according to the constraints $W_{ij}^{(v)} \geq 0$ and $\left(\mathbf{W}_i^{(v)}\right)^T \mathbf{1} = 1$. For individual $\mathbf{W}_i^{(v)}$, Problem (21) is reformulated as

$$\min_{\mathbf{W}_i^{(v)}} \left\|\mathbf{W}_i^{(v)} - \mathbf{C}_i^{(v)}\right\|_F^2 \ s.t.\ W_{ij}^{(v)} \geq 0,\ \left(\mathbf{W}_i^{(v)}\right)^T \mathbf{1} = 1 \tag{22}$$

where $\mathbf{C}_i^{(v)}$ is the ith column of $\mathbf{C}^{(v)}$. Each $\mathbf{W}_i^{(v)}$ in Problem (22) can be solved by the Euclidean projection method with guaranteed sparsity [38], where $\mathbf{C}_i^{(v)}$ is normalized by $\mathbf{C}_i^{(v)} \leftarrow normalize_{(0,1]}\left(\mathbf{C}_i^{(v)}\right)$. The number of nonzero elements in $\mathbf{W}^{(v)}$ is less than the number of nonzero elements in $\mathbf{C}^{(v)}$ due to the sparsity of $\mathbf{W}^{(v)}$. As a result, $\mathbf{W}^{(v)}$ is approximately closer to a strict block diagonal matrix than $\mathbf{C}^{(v)}$.

By seeking individual $\left\{\mathbf{W}^{(v)}\right\}_{v=1}^{n_v}$, we capture the consistent information across incomplete multiple views. Furthermore, we construct a normalized Laplacian matrix $\mathbf{L}^{(v)} \in \mathbb{R}^{n\times n}$ for the vth view according to Equation (1), i.e.,

$$\mathbf{L}^{(v)} = \mathbf{I}_n^{(v)} - \left(\mathbf{D}^{(v)}\right)^{-1/2}\mathbf{W}^{(v)}\left(\mathbf{D}^{(v)}\right)^{-1/2}. \tag{23}$$

Similar to Problem (2), the optimization problem of spectral embedding can be formulated as follows:

$$\min_{\mathbf{H}^{(v)}} tr\left(\left(\mathbf{H}^{(v)}\right)^T \mathbf{L}^{(v)}\mathbf{H}^{(v)}\right)\ s.t.\ \left(\mathbf{H}^{(v)}\right)^T \mathbf{H}^{(v)} = \mathbf{I}_k. \tag{24}$$

Finally, we can achieve individual low-dimensional embedding matrices $\left\{\mathbf{H}^{(v)} \in \mathbb{R}^{n\times k}\right\}_{v=1}^{n_v}$ by solving Problem (24). Algorithm 2 summarizes the complete optimization procedure for $\left\{\mathbf{H}^{(v)}\right\}_{v=1}^{n_v}$.

We explore the global structures of high-dimensional data by solving $\mathbf{Z}_c^{(v)}$ in Problem (17). Furthermore, the sparsity of solving $\mathbf{W}^{(v)}$ in Problem (22) is introduced to capture the local structures of high-dimensional data. The low-rank criterion and the sparsity constraint imposed on the adjacency matrices of the incomplete multiple views are significant for capturing the local and global structures of high-dimensional data. In Problem (24), $\mathbf{H}^{(v)}$ is a low-dimensional embedding matrix derived from $\mathbf{L}^{(v)}$, which is associated with $\mathbf{W}^{(v)}$. Therefore, the global and local structures of high-dimensional data are effectively explored simultaneously using Algorithm 2.

3.3 Low-Rank Tensor-Based Spectral Embedding Fusion

Multiple views usually provide complementary information. Given individual low-dimensional embedding matrices $\left\{\mathbf{H}^{(v)}\right\}_{v=1}^{n_v}$, we first present a multiview embedding matrix fusion model to achieve a consensus low-dimensional embedding matrix $\mathbf{F} \in \mathbb{R}^{n\times k}$. The fusion model explores the complementary information across multiple views and is formulated as

$$\begin{aligned} &\max_{\mathbf{F},\alpha_v} tr\left(\mathbf{F}^T \sum_{v=1}^{n_v} \alpha^{(v)}\mathbf{H}^{(v)}\right) \\ &s.t.\ \sum_{v=1}^{n_v}\left(\alpha^{(v)}\right)^2 = 1,\ \alpha^{(v)} \geq 0,\ \mathbf{F}^T\mathbf{F} = \mathbf{I}_k \end{aligned} \tag{25}$$

where $\alpha^{(v)}$ is an adaptive weight factor used to balance the significance of the vth view. Here, $\mathbf{F}$ shares the consensus information across different views, and it is considered a fused result obtained by aligning multiple low-dimensional embedding matrices.

According to Equation (5), $\overline{\mathbf{H}}^{(v)}\left(\overline{\mathbf{H}}^{(v)}\right)^T$ is a block diagonal matrix for the vth view, where $\overline{\mathbf{H}}^{(v)} = \mathbf{P}^{(v)}\mathbf{H}^{(v)}$. Each block submatrix contains elements that are all ones. Hence, $\overline{\mathbf{H}}^{(v)}\left(\overline{\mathbf{H}}^{(v)}\right)^T$ is also a low-rank matrix. Furthermore, we have

$$\overline{\mathbf{H}}^{(v)}\left(\overline{\mathbf{H}}^{(v)}\right)^T = \mathbf{P}^{(v)}\mathbf{H}^{(v)}\left(\mathbf{H}^{(v)}\right)^T\left(\mathbf{P}^{(v)}\right)^T \tag{26}$$

where $\mathbf{P}^{(v)} = diag\left[p_1^{(v)}, p_2^{(v)}, ..., p_n^{(v)}\right]$ is a diagonal matrix with elements $p_i^{(v)} = \frac{1}{\sqrt{\mathbf{h}_i^{(v)T}\mathbf{h}_i^{(v)}}}$. Consequently, $\mathbf{H}^{(v)}\left(\mathbf{H}^{(v)}\right)^T$ is a low-rank matrix containing k diagonal block submatrices. Intuitively, we stack $\left\{\mathbf{H}^{(v)}\left(\mathbf{H}^{(v)}\right)^T\right\}_{v=1}^{n_v}$ into a third-order tensor $\mathcal{H} \in$

$\mathbb{R}^{n \times n \times n_v}$, whose first frontal slices form an $n \times n$ low-rank matrix. The tensor $\mathcal{H}$ can be decomposed into a low-rank component and a sparse component. However, it is impossible to capture complementary information across multiple views since each individual low-rank matrix in the first frontal slices of $\mathcal{H}$ is considered for an individual view. Thus, we consider $\mathcal{T} = \left\{ \mathbf{H}^{(v)} \left(\mathbf{H}^{(v)} \right)^T \right\}_{v=1}^{n_v} \in \mathbb{R}^{n \times n_v \times n}$ as a surrogate for $\mathcal{H}$. The first frontal slices of $\mathcal{T}$ are $n \times n_v$ matrices. Hence, a low-rank tensor constraint is introduced to discover the high-level correlations among different views. From the perspective of low-dimensional embedding matrix fusion, the introduction of $\mathcal{T}$ is beneficial for capturing complementary information across multiple views.

For incomplete multiview data, we formulate a low-dimensional embedding matrix fusion model based on a low-rank tensor as follows:

$$\begin{aligned} &\min_{\mathbf{F}, \left\{ \mathbf{H}^{(v)} \right\}_{v=1}^{n_v}, \alpha^{(v)}} \|\mathcal{T}\|_* - \beta tr \left(\mathbf{F}^T \sum_{v=1}^{n_v} \alpha^{(v)} \mathbf{H}^{(v)} \right) \\ &s.t. \quad \sum_{v=1}^{n_v} \left(\alpha^{(v)} \right)^2 = 1, \ \alpha^{(v)} \geq 0, \\ &\qquad \mathbf{F}^T \mathbf{F} = \mathbf{I}_k, \ \left(\mathbf{H}^{(v)} \right)^T \mathbf{H}^{(v)} = \mathbf{I}_k \end{aligned} \tag{27}$$

where β is a tradeoff parameter. This model requires each group of embedding vectors in $\mathbf{H}^{(v)}$ to be aligned with the low-dimensional embedding vectors in $\mathbf{F}$.

3.4 Optimization Strategy

Problem (27) can be solved by an alternating direction method of multipliers (ADMM) framework [39]. We introduce an auxiliary tensor variable $\mathcal{G}$ into Problem (27) and consider the following equivalent problem

$$\begin{aligned} &\min_{\mathbf{F}, \left\{ \mathbf{H}^{(v)} \right\}_{v=1}^{n_v}, \alpha^{(v)}, \mathcal{G}} \|\mathcal{G}\|_* - \beta tr \left(\mathbf{F}^T \sum_{v=1}^{n_v} \alpha^{(v)} \mathbf{H}^{(v)} \right) \\ &s.t. \quad \sum_{v=1}^{n_v} \left(\alpha^{(v)} \right)^2 = 1, \ \alpha^{(v)} \geq 0, \mathcal{G} = \mathcal{T}, \\ &\qquad \mathbf{F}^T \mathbf{F} = \mathbf{I}_k, \ \left(\mathbf{H}^{(v)} \right)^T \mathbf{H}^{(v)} = \mathbf{I}_k. \end{aligned} \tag{28}$$

The augmented Lagrangian function in Problem (28) is

$$\begin{aligned} \mathcal{L} \left(\mathbf{F}, \left\{ \mathbf{H}^{(v)} \right\}_{v=1}^{n_v}, \alpha^{(v)}, \mathcal{G} \right) =& \|\mathcal{G}\|_* - \beta tr \left(\mathbf{F}^T \sum_{v=1}^{n_v} \alpha^{(v)} \mathbf{H}^{(v)} \right) \\ &+ \langle \mathcal{R}, \mathcal{T} - \mathcal{G} \rangle + \frac{\mu}{2} \|\mathcal{T} - \mathcal{G}\|_F^2 \end{aligned} \tag{29}$$

where $\mathcal{R}$ is a Lagrange multiplier, and $\mu > 0$ is an adaptive penalty parameter. The augmented Lagrangian function of Problem (29) can be transformed into the following equivalent function by using linear algebra techniques:

$$\begin{aligned} \mathcal{L} \left(\mathbf{F}, \left\{ \mathbf{H}^{(v)} \right\}_{v=1}^{n_v}, \alpha^{(v)}, \mathcal{G} \right) =& \|\mathcal{G}\|_* - \beta tr \left(\mathbf{F}^T \sum_{v=1}^{n_v} \alpha^{(v)} \mathbf{H}^{(v)} \right) \\ &+ \frac{\mu}{2} \left\| \mathcal{G} - \left(\mathcal{T} + \frac{\mathcal{R}}{\mu} \right) \right\|_F^2. \end{aligned} \tag{30}$$

The variables $\mathbf{F}$, $\left\{ \mathbf{H}^{(v)} \right\}_{v=1}^{n_v}$, $\alpha^{(v)}$ and $\mathcal{G}$ are updated alternately while the other variables are kept fixed until the algorithm converges. Problem (30) can be transformed into the four steps shown below. The updating scheme for the tth iteration is formulated as follows.

We first update $\mathbf{F}_t$ with fixed $\left\{ \mathbf{H}_{(t-1)}^{(v)} \right\}_{v=1}^{n_v}$, $\alpha_{(t-1)}^{(v)}$ and $\mathcal{G}_{(t-1)}$. Equation (30) can be rewritten as

$$\max_{\mathbf{F}_t} tr \left(\mathbf{F}_t^T \sum_{v=1}^{n_v} \alpha_{(t-1)}^{(v)} \mathbf{H}_{(t-1)}^{(v)} \right) \quad s.t. \quad \mathbf{F}_t^T \mathbf{F}_t = \mathbf{I}_k. \tag{31}$$

Let $\mathbf{H}_F = \sum_{v=1}^{n_v} \alpha_{(t-1)}^{(v)} \mathbf{H}_{(t-1)}^{(v)}$; then, and Problem (31) has a closed-form solution [40], i.e.,

$$\mathbf{F}_t = \mathbf{U}_F \mathbf{V}_F^T \tag{32}$$

where the SVD of $\mathbf{H}_F$ is $\mathbf{H}_F = \mathbf{U}_F \mathbf{\Sigma}_F \mathbf{V}_F^T$. Here, $\mathbf{V}_F$ is an orthogonal matrix, i.e., $\mathbf{V}_F \left(\mathbf{V}_F \right)^T = \left(\mathbf{V}_F \right)^T \mathbf{V}_F = \mathbf{I}$. Hence, $\mathbf{V}_F$ can be regarded as a rotation matrix for $\mathbf{F}_t$, which makes $\mathbf{F}_t$ approximate to the discrete indicator matrix.

Then, we update $\left\{ \mathbf{H}_t^{(v)} \right\}_{v=1}^{n_v}$ with fixed $\mathbf{F}_t$, $\alpha_{(t-1)}^{(v)}$ and $\mathcal{G}_{(t-1)}$. Equation (30) can be reformulated as

$$\begin{aligned} &\min_{\left\{ \mathbf{H}_t^{(v)} \right\}_{v=1}^{n_v}} -\beta tr \left(\mathbf{F}_t^T \sum_{v=1}^{n_v} \alpha_{(t-1)}^{(v)} \mathbf{H}_t^{(v)} \right) \\ &+ \frac{\mu}{2} \left\| \mathcal{G}_{(t-1)} - \left(\mathcal{T}_t + \frac{\mathcal{R}_{(t-1)}}{\mu_{(t-1)}} \right) \right\|_F^2 \\ &s.t. \quad \left(\mathbf{H}_t^{(v)} \right)^T \mathbf{H}_t^{(v)} = \mathbf{I}_k. \end{aligned} \tag{33}$$

We stack $\left\{ \mathbf{H}_t^{(v)} \left(\mathbf{H}_t^{(v)} \right)^T \right\}_{v=1}^{n_v}$ into a third-order tensor $\mathcal{T}_t$, whose first frontal slices form a matrix of size $n \times n_v$. Let $\mathcal{A} = \mathcal{G}_{(t-1)} - \frac{\mathcal{R}_{(t-1)}}{\mu_{(t-1)}}$ and $\mathcal{A}^{(v)}$ is the vth frontal slice of $\mathcal{A}$. For ease of computation, we obtain the relaxed problem

$$\begin{aligned} &\min_{\mathbf{H}_t^{(v)}} tr \left(\left(\mathbf{H}_t^{(v)} \right)^T \mathbf{B}^{(v)} \right) + tr \left(\left(\mathbf{H}_t^{(v)} \right)^T \mathbf{C}^{(v)} \mathbf{H}_t^{(v)} \right) \\ &s.t. \quad \left(\mathbf{H}_t^{(v)} \right)^T \mathbf{H}_t^{(v)} = \mathbf{I}_k \end{aligned} \tag{34}$$

where $\mathbf{B}^{(v)} = -\beta \alpha_{(t-1)}^{(v)} \mathbf{F}_t$ and $\mathbf{C}^{(v)} = \frac{\mu_{(t-1)}}{2} \left(\mathbf{H}_{(t-1)}^{(v)} \left(\mathbf{H}_{(t-1)}^{(v)} \right)^T - \left(\mathcal{A}^{(v)} + \left(\mathcal{A}^{(v)} \right)^T \right) \right)$. Problem (34) can be solved by a first-order framework [41].

Next, we update $\mathcal{G}_t$ with fixed $\mathbf{F}_t$, $\left\{ \mathbf{H}_t^{(v)} \right\}_{v=1}^{n_v}$ and $\alpha_{(t-1)}^{(v)}$. The solution to $\mathcal{G}_t$ can be obtained by solving the following problem:

$$\min_{\mathcal{G}_t} \|\mathcal{G}_t\|_* + \frac{\mu}{2} \left\| \mathcal{G}_t - \left(\mathcal{T}_t + \frac{\mathcal{R}_{(t-1)}}{\mu_{(t-1)}} \right) \right\|_F^2. \tag{35}$$

Problem (35) can be solved via the t-SVT operator [31], which is given in Algorithm 1.

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2022.3230964

Algorithm 3 Solving (27) by using an ADMM framework

Input: The low-dimensional embedding matrices $\left\{\mathbf{H}^{(v)}\right\}_{v=1}^{n_v}$ and a parameter $\beta > 0$.

Initialize: $\alpha_0^{(v)} = \sqrt{\frac{1}{n_v}}$, $\mathbf{H}_0^{(v)} = \mathbf{H}^{(v)}$, $\mathcal{G}_0 = \mathcal{R}_0 = 0$, $\rho = 1.2$, $\mu = 10^{-4}$, $\mu_0 = \mu$, $\mu_{\max} = 10^6$, $\varepsilon = 10^{-6}$, $t = 1$ and $maxIters = 100$.

while not converged **do**
  update $\mathbf{F}_t$ by using (32);
  update $\left\{\mathbf{H}_t^{(v)}\right\}_{v=1}^{n_v}$ by using a first-order framework [41];
  update $\mathcal{G}_t$ by using Algorithm 1;
  update $\left\{\alpha_t^{(v)}\right\}_{v=1}^{n_v}$ by using (37);
  update the Lagrange multiplier $\mathcal{R}_t$ by using (38);
  update the parameter μ_t by using (39);
  check the convergence condition:
  $\|\mathcal{G}_t - \mathcal{T}_t\|_{\max} < \varepsilon$;
  if $t < maxIters$ and $not\ converged$ **then**
    $t \leftarrow t + 1$;
  else
    break;
  end if
end while

Output: $\mathbf{F}_t$.

Finally, we update $\alpha_t^{(v)}$ with fixed $\mathbf{F}_t$, $\left\{\mathbf{H}_t^{(v)}\right\}_{v=1}^{n_v}$ and $\mathcal{G}_t$. Equation. (30) can be rewritten as

$$\begin{aligned} &\max_{\left\{\alpha_t^{(v)}\right\}_{v=1}^{n_v}} tr\left(\mathbf{F}_t^T \sum_{v=1}^{n_v} \alpha_t^{(v)} \mathbf{H}_t^{(v)}\right) \\ &s.t. \quad \sum_{v=1}^{n_v} \left(\alpha_t^{(v)}\right)^2 = 1,\ \alpha_t^{(v)} \geq 0. \end{aligned} \tag{36}$$

The optimal solution to Problem (36) is given by:

$$\alpha_t^{(v)} = \frac{\mathbf{H}_\alpha^{(v)}}{\sqrt{\sum_{i=1}^{n_v} \left(\mathbf{H}_\alpha^{(i)}\right)^2}} \tag{37}$$

where $\mathbf{H}_\alpha^{(v)} = tr\left(\mathbf{F}_t^T \mathbf{H}_t^{(v)}\right)$ [42].

In addition, $\mathcal{R}_t$ and $\mu_{(t-1)}$ are updated during the tth iteration. Given $\mathcal{G}_t$, $\mathcal{T}_t$ and μ_k, the Lagrange multiplier $\mathcal{R}_t$ is updated as

$$\mathcal{R}_t = \mathcal{R}_{(t-1)} + \mu_{(t-1)} \left(\mathcal{G}_t - \mathcal{T}_t\right). \tag{38}$$

The penalty parameter μ_t is updated as

$$\mu_t = \min(\rho \mu_{(t-1)}, \mu_{\max}) \tag{39}$$

where ρ and $\mu_{\max}$ are constants. These steps are performed repeatedly until the convergence condition is satisfied, i.e., $\|\mathcal{G}_{t+1} - \mathcal{T}_{t+1}\|_{\max} < \varepsilon$, or the number of iterations exceeds a maximum threshold. Algorithm 3 summarizes the entire procedure of the approach for optimizing Problem (27).

3.5 Theoretical Analysis of LRTL

After obtaining the consensus low-dimensional embedding matrix $\mathbf{F}$, we execute the k-means algorithm [30] on $\mathbf{F}$ to

Algorithm 4 The LRTL algorithm

Input: Data matrices $\left\{\mathbf{X}_c^{(v)}\right\}_{v=1}^{n_v}$, diagonal indicator matrices $\left\{\mathbf{M}^{(v)}\right\}_{v=1}^{n_v}$, parameters $\lambda > 0$, $\beta > 0$ and the number of clusters k.

for $v = 1$ to n_v **do**
  $\mathbf{X}_c^{(v)}$ consists of the nonzero columns of $\mathbf{X}^{(v)}\mathbf{M}^{(v)}$;
end for
Obtaining the low-dimensional embedding matrices $\left\{\mathbf{H}^{(v)}\right\}_{v=1}^{n_v}$ by using Algorithm 2;
Obtaining $\mathbf{F}$ by using Algorithm 3;
Executing the k-means algorithm on $\mathbf{F}$ to obtain k clusters [30];

Output: The k clusters.

obtain k clusters. The complete IMVC procedure is outlined in Algorithm 4. Each diagonal indicator matrix $\mathbf{M}^{(v)}$ becomes an identity matrix of size $n \times n$ in Equation (13) if all instances are available in multiple views. Thus, the mapping function (20) is redundant since it degenerates into the equivalent mapping function. This indicates that the proposed LRTL method can also be applied for MVC.

3.5.1 An Alternative Fusion Strategy

We present an alternative fusion strategy to improve the efficiency of LRTL in the fusion model. Considering $\mathbf{B}^{(v)}$ and $\mathbf{C}^{(v)}$ in Problem (34), we suppose that

$$\left\|\mathbf{C}^{(v)}\right\|_{\max} \Big/ \left\|\mathbf{B}^{(v)}\right\|_{\max} \leq \delta \tag{40}$$

where $\|\cdot\|_{\max}$ represents the maximum absolute value among all the elements in a matrix and δ is a small positive value, e.g., $\delta = 1e^{-6}$. This implies that $\mathbf{H}^{(v)}$ is heavily dependent on $\mathbf{F}$ in Problem (34). The dimensions of the third-order tensors $\mathcal{T}$ and $\mathcal{G}$ corresponding to the first frontal slices are reduced from n_v to 1 in Problem (35) because all $\left\{\mathbf{H}^{(v)}\right\}_{v=1}^{n_v}$ are identical for $v \in [1, n_v]$. Moreover, $\left\{\alpha_t^{(v)}\right\}_{v=1}^{n_v}$ are also identical according to Equation (37). This indicates that $\left\{\alpha_t^{(v)}\right\}_{v=1}^{n_v}$ can be removed from Problem (27). Therefore, updating $\left\{\mathbf{H}_t^{(v)}\right\}_{v=1}^{n_v}$ can be relaxed to finding any individual $\mathbf{H}_t^{(v)}$ in Problem (34) if we set $\mu = 10^{-4}$ in Algorithm 3 as along with a proper β. After the first iteration, $\mathbf{F}$ is equal to any individual $\mathbf{H}_t^{(v)}$ in Problem (31). The above consensus analysis on $\left\{\mathbf{H}^{(v)}\right\}_{v=1}^{n_v}$ and $\mathbf{F}$ explains why the fusion model is able to work if condition (40) imposed on the values of μ and β is satisfied in Algorithm 3.

In Algorithm 3, the t-SVT operator involves SVD operations for n matrices of size $n \times n_v$. This results in a relatively heavy computational cost. Since the sizes of the first frontal slices of $\mathcal{T}$ and $\mathcal{G}$ are reduced from $n \times n_v$ to $n \times 1$, we can avoid these SVD operations. Suppose that $\mathbf{T} \in \mathbb{R}^{n \times 1}$ is a matrix of size $n \times 1$. Let $\mathbf{T} \neq 0$ and the economy SVD of $\mathbf{T}$ is given by

$$\mathbf{T} = \mathbf{U}_T \Sigma_T \tag{41}$$

where $\mathbf{U}_T = \mathbf{T}/\|\mathbf{T}\|_2 \in \mathbb{R}^{n\times 1}$ and $\Sigma_T = \|\mathbf{T}\|_2 \in \mathbb{R}^{1\times 1}$. Each SVD operation can be replaced by Equation (41) when we apply the t-SVT operator in Algorithm 1 [31] to solve $\mathcal{G}$ in Problem (35). Such a replacement is able to effectively reduce the computational cost of the technique. Consequently, this alignment model improvement is considered a surrogate for updating $\left\{\mathbf{H}_t^{(v)}\right\}_{v=1}^{n_v}$ in Algorithm 3.

3.5.2 Convergence Analysis

The general convergence properties of the ADMM framework have been investigated in theory. Algorithm 4 performs well in practical applications. We provide an explanation for the convergence condition $\|\mathcal{G}_{t+1} - \mathcal{T}_{t+1}\|_{\max} < \varepsilon$ of Algorithm 4. According to Equation (39), μ dramatically rises with $\rho > 1$ as t steadily increases. This implies that $\mathcal{G}$ approaches to $\mathcal{T}$ in Problem (35) as $\mu \to \infty$, where all $\left\{\mathbf{H}^{(v)}\right\}_{v=1}^{n_v}$ are identical in $\mathcal{T}$. Consequently, the convergence condition will eventually be satisfied as t constantly increases under certain conditions, i.e., the appropriate initialization values of μ and ρ.

3.5.3 Computational Complexity Analysis

We first consider the computational complexity for an individual view. The computational complexities of low-rank and sparsity computations are $O\left(d_v N_v^2 + N_v^2\right)$ and $O\left(d_v \log\left(d_v\right) n\right)$, respectively. The computational complexities of the construction process and the eigenvalue decomposition of the normalized Laplacian matrix $\mathbf{L}^{(v)}$ are $O\left(n^2\right)$ and $O\left(n^3\right)$, respectively. Hence, the total computational complexity of Algorithm 2 is $O\left(\sum_{v=1}^{n_v}\left(d_v N_v^2 + d_v \log\left(d_v\right) n\right) + n_v n^3\right)$. In Algorithm 3, three important variables $\mathbf{F}$, $\mathbf{H}^{(v)}$ and $\mathcal{G}$ are updated alternately during each iteration. The first step of Algorithm 3 that updates $\mathbf{F}$ requires the SVD of the matrix $\mathbf{H}^{(v)}$, whose computational complexity is $O(kn^2)$. The second and third steps of Algorithm 3 that update $\mathbf{H}^{(v)}$ and $\mathcal{G}$ require computational complexities of $O(t_1 kn^2)$ and $O(n^2)$, respectively, where t_1 is the number of iterations needed to solve Problem (34). In addition, k-means requires a computational complexity of $O(t_2 k^2 n)$ in Algorithm 4, where t_2 is the number of iterations. Therefore, the final overall complexity of Algorithm 4 is $O\left(\sum_{v=1}^{n_v} d_v N_v^2 + n_v n^3 + k t_1 t_3 n^2\right)$, where $d_v \ll n$, $k \ll n$, $t_1 \ll n$, $t_2 \ll n$, $t_3 \ll n$ and t_3 is the number of iterations required for solving Problem (27).

4 Experiments

In this section, we conduct a series of experiments to evaluate the effectiveness of the proposed LRTL method on benchmark datasets. The source code for the proposed method is implemented in MATLAB 2021b, and is available online[1]. All experiments are conducted on a Windows 10 platform with an Intel i7-10700 CPU and 32 GB of RAM.

TABLE 2
Statistics of the datasets.

Dataset	Clusters	Views	Data samples
Reuters	6	5	600
O-Scene	8	4	2688
Handwritten	10	6	2000
Flower17	17	7	1360
COIL-20	20	3	1440
ProteinFold	27	12	694
SUN RGB-D	45	2	10,335
100leaves	100	3	1600
Caltech101	101	6	8677

4.1 Experimental Settings

4.1.1 Datasets

Nine multiview benchmark datasets are used to evaluate the proposed LRTL method in the experiments. The statistics of the datasets are summarized in Table 2. The descriptions of the datasets are listed as follows.

- **Reuters Dataset** [43]: This dataset contains 600 documents written in five languages and their translations over a common set of six categories. We randomly select 600 documents for this dataset, where each class contains 100 documents.
- **Outdoor Scene (O-Scene) Dataset** [44]: This dataset has 2688 images consisting of 8 groups. For each image, we extract four different feature vectors.
- **Handwritten Dataset** [45]: This dataset consists of 2,000 images of ten handwritten digits (0–9), each of which is represented by six different features.
- **ProteinFold Dataset** [2]: This dataset contains 694 protein domains that belong to 27 classes. Each protein domain is represented by 12 views.
- **Flower17 Dataset** [46]: This dataset includes 17 different flower categories, where each class has 80 images. Each image is represented by 7 views.
- **COIL-20 Dataset** [47]: This dataset is composed of 1,440 images of 20 objects in which the background has been discarded. Each image is represented by three kinds of features, including a 1024-dimension intensity, 3304-dimension local binary pattern and 6750-dimension Gabor.
- **SUN RGB-D Dataset** [48]: This dataset has 10,335 RGB-D images. The features are extracted from the original images using the deep neural network [49].
- **100leaves Dataset** [50]: This dataset contains 1,600 samples of 100 categories. The shape descriptor, fine scale margin and texture histogram features are extracted to depict each sample.
- **Caltech-101 Dataset** [51]: This dataset contains 8,677 images of objects that belong to 101 classes, where we remove the background category. Each object has approximately 30-800 images.

4.1.2 Comparison Methods

We compare our approach with five state-of-the-art methods, the descriptions of which are given as follows:

- **DPSC** [28]: The distribution preserving subspace clustering (DPSC) algorithm constructs a latent distribution-preserving autoencoder.

1. https://codeocean.com/capsule/3481358/tree/v1

2. http://mkl.ucsd.edu/dataset/protein-fold-prediction

TABLE 3
Average clustering results and standard deviations of the different methods on nine multiview datasets with various missing ratios.

Datasets	Methods	ACC				NMI				F-measure			
		0	0.1	0.3	0.5	0	0.1	0.3	0.5	0	0.1	0.3	0.5
Reuters	$\mathrm{LRTL_{Agg}}$	51.07±0.16	49.33±0.00	48.98±0.05	42.17±0.00	34.68±0.09	33±0.00	25.47±0.01	23.57±0.00	52.49±0.16	49.67±0.00	48.47±0.04	45.35±0.00
	DPSC	55.17±0.00	56.33±0.00	49.33±0.00	45.17±0.00	37.6±0.00	37.14±0.00	35.7±0.00	29.75±0.00	55.42±0.00	56.63±0.00	51.54±0.00	48.41±0.00
	SRSC	86.75±1.72	86.76±1.34	70.37±6.29	64.11±6.85	77.77±1.64	77.56±1.7	68.39±3.7	58.59±4.68	86.87±1.72	86.83±1.31	73.38±5.5	67±5.97
	GIMC	50.83±0.21	48.5±0.00	48±0.00	49.17±0.00	29.24±0.17	32.83±0.12	25.99±0.00	36.23±0.00	52.65±0.19	49.36±0.11	47.29±0.00	51.45±0.00
	EE-IMVC	52±0.00	52.17±0.00	52±0.00	53.5±0.00	34.48±0.00	34.97±0.00	35.11±0.00	36.08±0.00	51.71±0.00	51.48±0.00	51.85±0.00	53.59±0.00
	TMBDSD	76.67±0.00	74.97±0.11	77.62±0.11	78.5±0.24	57.89±0.00	55.77±0.16	62.82±0.28	65.83±0.31	76.61±0.00	74.99±0.01	77.18±0.01	78.28±0.27
	LRTL	**96.5±0.25**	**92.83±0.00**	**98.33±0.00**	**94.78±0.97**	**91.88±0.58**	**85.91±0.00**	**95.36±0.00**	**88.66±1.56**	**96.5±0.25**	**92.83±0.00**	**98.33±0.00**	**94.78±0.97**
O-Scene	$\mathrm{LRTL_{Agg}}$	54.61±0.32	57.56±0.12	44.42±0.04	28.02±0.17	40.6±0.19	42.44±0.11	27.26±0.07	16.54±0.29	53.47±0.32	56.93±0.13	46.38±0.03	33.26±0.29
	DPSC	62.17±0.00	61.05±0.00	52.9±0.00	44.12±0.00	50.09±0.00	46.39±0.00	36.53±0.00	29.02±0.00	61.62±0.00	60.13±0.00	52.95±0.00	46.95±0.00
	SRSC	73.65±0.02	65.74±0.02	57.22±0.01	37.37±0.05	59.29±0.02	49.65±0.02	40.68±0.01	22.87±0.05	73.23±0.02	65.43±0.02	56.98±0.01	40.96±0.06
	GIMC	51.76±0.03	51.7±0.04	41.11±0.57	32.55±0.25	35.54±0.03	35.46±0.02	26.55±0.31	23.52±0.16	51.03±0.05	50.95±0.05	42.27±0.56	36.02±0.24
	EE-IMVC	64.17±0.2	64.83±0.26	59.18±0.23	46.87±0.16	49.16±0.21	48.74±0.19	41.37±0.08	29.44±0.12	63.89±0.19	64.29±0.26	59.07±0.23	48.04±0.1
	TMBDSD	70.8±0.04	41.35±0.08	38.26±0.03	30.13±0.09	57.79±0.06	29.03±0.07	25.19±0.05	14.47±0.09	70.04±0.05	41.19±0.07	38.42±0.04	32.64±0.07
	LRTL	**96.09±0.00**	**95.65±0.00**	**94.4±0.02**	**87.25±0.07**	**91.8±0.00**	**89.91±0.00**	**89.17±0.03**	**76.47±0.11**	**96.08±0.00**	**95.65±0.00**	**94.4±0.02**	**87.25±0.07**
Handwritten	$\mathrm{LRTL_{Agg}}$	86.22±0.03	73±0.02	60.67±0.04	39.67±0.12	81±0.05	72.86±0.02	51.39±0.14	39.15±0.17	86.1±0.3	76.38±0.01	65.49±0.11	44.58±0.19
	DPSC	88.15±0.00	87.6±0.00	79.4±0.00	74.45±0.00	79.67±0.00	79.43±0.00	73.82±0.00	65.69±0.00	88.17±0.00	87.65±0.00	81.26±0.00	76.46±0.00
	SRSC	92.02±0.04	92.15±0.00	91.14±0.02	82.41±0.04	85.04±0.09	85.38±0.00	83.23±0.06	69.74±0.04	92.03±0.04	92.17±0.00	91.16±0.02	82.51±0.04
	GIMC	87.72±0.05	84.17±0.06	85.32±0.22	77.01±0.14	78.96±0.05	77.58±0.11	77.99±0.19	69.38±0.08	87.7±0.05	84.15±0.07	85.35±0.23	77.01±0.15
	EE-IMVC	88.53±0.14	89.49±0.06	89.84±0.06	82.84±0.06	79.72±0.14	80.79±0.09	79.75±0.09	69.42±0.12	88.55±0.15	89.48±0.05	89.89±0.06	82.9±0.06
	TMBDSD	99.5±0.00	97.7±0.00	94.05±0.00	94.68±0.03	98.81±0.00	94.89±0.00	91.04±0.00	87.85±0.05	99.5±0.00	97.71±0.00	93.98±0.00	94.68±0.03
	LRTL	**99.59±0.02**	**99.27±0.05**	**98.8±0.00**	**98.15±0.00**	**98.92±0.04**	**98.28±0.13**	**97.2±0.00**	**95.98±0.00**	**99.59±0.02**	**99.27±0.05**	**98.8±0.00**	**98.15±0.00**
Flower17	$\mathrm{LRTL_{Agg}}$	61.01±0.2	59.2±0.12	55.82±0.13	39.28±0.47	57.34±0.15	56.72±0.16	50.52±0.18	40.17±0.42	64.01±0.2	62.39±0.14	57.68±0.15	44.79±0.5
	DPSC	47.87±0.00	48.31±0.00	43.53±0.00	37.94±0.00	47.76±0.00	47.16±0.00	43.21±0.00	34.74±0.00	49.72±0.00	51.63±0.00	45.98±0.00	40.87±0.00
	SRSC	48.37±0.59	46.27±0.29	45.4±0.62	37.58±0.3	46.55±0.62	47.49±0.4	43.1±0.43	37.7±0.27	51.56±0.57	51.82±0.35	49.23±0.69	42.44±0.38
	GIMC	45.71±0.5	44.85±1.07	43.38±1.08	38.93±0.71	49.41±0.38	45.83±0.46	43.16±0.35	40.42±0.26	49.67±0.35	46.94±0.64	45.2±0.5	40.69±0.34
	EE-IMVC	59.77±0.49	57.38±0.36	54.91±0.61	49.35±1.4	57.11±0.36	54.4±0.4	50.67±0.38	43.49±0.85	61.79±0.29	60.49±0.36	56.73±0.43	50.93±1.33
	TMBDSD	76.4±0.28	88.42±0.28	86.32±0.09	87.97±0.14	74.41±0.23	86.93±0.14	83.57±0.15	89.55±0.06	78.11±0.27	88.33±0.27	85.93±0.09	88.28±0.05
	LRTL	**96.13±0.54**	**92.88±0.16**	**91.63±0.71**	**91.13±0.12**	**94.43±0.63**	**91.5±0.09**	**90.85±0.55**	**89.18±0.09**	**96.13±0.54**	**92.83±0.17**	**91.64±0.71**	**91.1±0.12**
COIL-20	$\mathrm{LRTL_{Agg}}$	72.98±0.02	73.13±0.04	72.89±0.68	59.51±1.89	81.43±0.05	80.01±0.06	79.72±0.24	67.73±0.7	74.48±0.03	73.2±0.04	74.75±0.27	64.93±1.66
	DPSC	76.04±0.00	73.13±0.00	69.03±0.00	60.9±0.00	83.24±0.00	83.68±0.00	76.98±0.00	69.04±0.00	76.49±0.00	76.13±0.00	71.2±0.00	67.45±0.00
	SRSC	79.31±0.97	76.97±3.02	80.01±0.57	69.12±0.9	86±0.24	85.71±1.32	84.34±0.41	74.6±0.79	81.54±0.94	79.39±2.24	81.71±0.37	74.36±0.98
	GIMC	75.96±1.2	73.65±1.35	72.32±1.01	58.13±0.76	84.48±0.88	81.19±0.73	79.66±0.86	66.42±0.7	78.12±0.13	74.3±0.98	73.25±1.16	61.08±0.71
	EE-IMVC	74.15±1.02	73.75±0.99	71.93±0.5	64.72±0.31	82.73±0.57	82.02±1.06	77.86±0.53	69.51±0.26	75.03±0.44	74.53±1.03	73.48±0.55	69.89±0.28
	TMBDSD	90.09±0.03	87.67±0.05	89.74±0.07	90.28±0.47	91.33±0.06	89.53±0.03	90.52±0.1	89.51±0.35	90.16±0.02	87.61±0.05	89.78±0.06	90.21±0.46
	LRTL	**95.9±0.88**	**89.83±0.23**	**95.06±0.83**	**93.36±0.59**	**94.53±0.96**	**90.71±0.18**	**93.88±0.79**	**92.01±0.51**	**95.9±0.88**	**89.84±0.23**	**95.06±0.83**	**93.36±0.59**
ProteinFold	$\mathrm{LRTL_{Agg}}$	34.01±1.22	33.03±1.24	27.67±0.46	21.08±0.6	43.29±0.9	41.37±0.88	35.31±0.41	28.36±0.44	36.27±1.52	35.71±1.22	30.34±0.5	23.85±0.51
	DPSC	29.25±0.00	30.98±0.00	29.54±0.00	28.24±0.00	39.68±0.00	39.59±0.00	39.93±0.00	35.36±0.00	32.03±0.00	34.07±0.00	32.7±0.00	31.09±0.00
	SRSC	37.71±1.31	36.08±0.8	34.25±1.54	31.64±0.92	46.62±1.1	44.1±0.66	43.43±1	39.47±0.4	39.77±1.12	38.01±0.83	36.46±1.42	34.02±0.98
	GIMC	23.8±0.78	22.95±0.96	20.12±1.08	17.32±0.67	34.05±0.62	32.78±0.69	28.62±1.16	23.43±0.51	26.78±0.65	26.22±1.09	23.28±0.9	19.49±0.56
	EE-IMVC	33.86±1.11	34.35±1.65	33.23±2.13	30.43±0.69	43.87±0.96	43.63±0.78	42.38±1.21	38.2±0.74	36.63±1.23	36.59±1.51	35.34±2.1	32.48±0.79
	TMBDSD	58.14±2.23	36.21±0.9	41.61±1.47	36.53±1.02	74.44±1.69	53.38±0.82	63.3±0.96	55.91±1.13	63.24±2.39	39.35±0.68	44.29±1.47	39.92±1.36
	LRTL	**59.32±1.5**	**55.88±2.56**	**51.97±1.86**	**45.13±1.39**	**78.29±1.01**	**75.55±0.99**	**71.17±1.29**	**69.45±0.85**	**64.35±1.05**	**59.8±2.05**	**56.08±2.41**	**50.11±1.31**
SUN RGB-D	$\mathrm{LRTL_{Agg}}$	18.69±0.38	17.45±0.21	14.53±0.35	12.75±0.31	29.27±0.21	25.96±0.15	21.1±0.31	16.69±0.18	25.13±0.27	23.77±0.3	20.39±0.47	17.62±0.49
	DPSC	20.25±0.51	19.61±1.04	17.63±0.58	15.72±0.51	29.92±0.24	26.75±0.38	20.72±0.37	14.95±0.3	27.38±0.57	26.13±1.22	23.87±0.65	19.91±0.6
	SRSC	19.45±0.27	18.84±0.51	16.13±0.3	13.04±0.45	29.66±0.19	25.93±0.21	21.04±0.15	16.53±023	25.84±0.46	25.5±0.72	22.51±0.4	17.5±0.59
	GIMC	16.46±0.42	15.31±0.41	14.43±0.13	13.68±0.15	23.61±0.65	23.84±0.18	22.46±0.32	19.28±0.26	22.91±0.14	21.19±0.25	19.89±0.24	15.92±0.37
	EE-IMVC	19.76±0.7	19.46±0.65	16.25±0.46	12.83±0.49	27.09±0.28	24.23±0.33	18.68±0.28	14.6±0.27	26.45±0.82	26.2±0.75	24.44±0.61	22.42±0.43
	TMBDSD	19.79±0.38	17.47±0.73	16.98±0.44	16.29±0.32	28.64±0.38	23.5±0.48	22.66±0.2	19.95±0.03	26.49±0.51	24.29±1.01	22.43±0.46	18.36±1.24
	LRTL	**23.76±0.88**	**22.48±0.75**	**22.05±0.67**	**22.29±0.36**	**39.17±0.49**	**37.57±0.6**	**38.05±0.34**	**36.01±0.28**	**30.69±1.01**	**29.17±0.8**	**29.39±0.64**	**29.43±0.28**
100leaves	$\mathrm{LRTL_{Agg}}$	77.2±0.8	69.01±1.53	47.61±0.99	35.09±0.81	88.78±0.06	83.3±0.82	70.72±0.29	61.21±0.28	79.61±0.92	71.67±1.38	51.86±0.6	40.54±0.61
	DPSC	80.69±0.00	66±0.00	48.13±0.00	33.5±0.00	91.72±0.00	83.48±0.00	70.64±0.00	59.94±0.00	82.15±0.00	69.96±0.00	52.57±0.00	38.81±0.00
	SRSC	82.42±1.52	71.24±0.94	51.37±1.53	34.48±0.56	92.56±0.51	85.06±0.48	72.62±0.64	61.45±0.37	84.64±1.33	74.47±0.82	56.45±1.28	39.59±0.77
	GIMC	75.3±1.25	73.04±1.74	47.13±1.99	30.69±1.73	91.08±0.47	89.73±0.32	73.81±0.58	58.86±0.59	79.25±1.14	77.47±1.14	50.27±1.65	33.7±1.36
	EE-IMVC	74.17±1.81	63.56±1.36	43.2±0.75	30.62±0.61	87.76±0.55	80.61±0.51	68.04±0.55	58.57±0.67	76.59±1.32	66.5±1.16	47.68±0.76	35.15±0.65
	TMBDSD	90.83±1.24	90.92±1.09	88.79±0.89	87.48±1.09	96.67±0.24	96.34±0.29	95.01±0.18	93.06±0.26	93.08±0.82	92.98±0.77	91.04±0.59	89.52±0.83
	LRTL	**92.49±0.87**	**92.55±0.88**	**91.1±0.95**	**88.56±1.00**	**98.3±0.21**	**98.13±0.27**	**97.25±0.3**	**95.25±0.41**	**94.65±0.67**	**94.52±0.66**	**93.11±0.78**	**90.41±0.92**
Caltech-101	$\mathrm{LRTL_{Agg}}$	20.92±0.46	20.7±0.5	16.29±0.23	12.91±0.19	45.39±0.26	43.64±0.21	36.16±0.19	31.67±0.3	26.14±0.44	26.01±0.78	21.01±0.29	16.49±0.25
	DPSC	27.68±0.00	26.47±0.00	25.84±0.00	24.2±0.00	51.71±0.00	50.81±0.00	46.52±0.00	42.51±0.00	35.27±0.00	33.63±0.00	31.46±0.00	30.28±0.00
	SRSC	23.73±0.53	20±0.4	21.9 ±0.53	18.27±0.55	48.47±0.38	42.92±0.21	42.26±0.21	37.01±0.41	29.99±0.57	26.39±0.61	27.65±0.52	23.01±0.69
	GIMC	18.11±0.00	16.09±0.00	15.02±0.00	13.14±0.00	39.05±0.00	36.02±0.00	35.1±0.00	30.52±0.00	23±0.00	19.61±0.00	18.24±0.00	16.18±0.00
	EE-IMVC	27.12±0.88	26.57±0.57	21.36±0.92	18.9±0.55	51.67±0.33	50.16±0.34	43.56±0.19	36.76±0.34	33.21±1.11	31.61±0.61	27.41±0.87	23.4±0.57
	TMBDSD	32.38±1.18	32.75±0.46	29.14±0.9	24.56±0.23	59.14±0.58	59.08±0.66	54.29±0.19	47.5±0.41	38.04±1.07	38.26±0.24	35.32±1.21	30.68±0.53
	LRTL	**60.86±1.26**	**60.04±1.02**	**60.81±1.73**	**59.05±1.84**	**87.05±0.41**	**87.27±0.39**	**87.1±0.5**	**87.14±0.48**	**67.05±1.23**	**65.88±1.24**	**66.45±153**	**65.36±1.74**

- **SRSC** [52]: The self-representation subspace clustering (SRSC) algorithm jointly performs data imputation and self-representation learning.
- **GIMC** [53]: The generalized incomplete multiview clustering (GIMC) algorithm considers the local geometric information and the unbalanced discriminant information of incomplete multiple views.
- **EE-IMVC** [16]: The efficient and effective incomplete multiview clustering (EE-IMVC) algorithm imputes each incomplete base matrix generated by incomplete multiple views with a learned consensus clustering matrix.
- **TMBDSD** [17]: The tensor-based multiview block diagonal structure diffusion (TMBDSD) algorithm explores block-diagonal structures of incomplete multiview data with a tensor low-rank constraint.

The source codes of the competing algorithms are provided by their authors. In addition, we consider a variant of the proposed LRTL method, namely, LRTL_{Agg}, to validate the effectiveness of the spectral embedding-based fusion of LRTL for IMVC. Specifically, we aggregate all similarity

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2022.3230964

TABLE 4
Computational times (in seconds) of the different methods on nine multiview datasets with various missing ratios.

Datasets	Ratio	$\text{LRTL}_{\textbf{Agg}}$	DPSC	SRSC	GIMC	EE-IMVC	TMBDSD	LRTL
Reuters	0	3.3	3.9	1.1	6.3	**1.1**	21.3	5.6 (7.8)
	0.1	2.5	3.6	1.4	5.3	**1.2**	16.1	5.8 (6)
	0.3	2.4	4.8	1.4	7.4	**1.2**	13.3	4.3 (6.1)
	0.5	2	3.5	1.2	5.7	**1.3**	12.9	4.9 (9.6)
O-Scene	0	32.2	53.2	19.8	149.6	**9.2**	499.1	111.8 (145.1)
	0.1	28.7	50.8	45.4	143.3	**10.3**	455.1	112.8 (142.3)
	0.3	13.3	51.7	52.5	137.6	**10.6**	384.2	102.6 (136.8)
	0.5	15	50.5	61.3	135.8	**12.9**	364.7	100.5 (132.2)
Handwritten	0	15.3	23	12.3	103.2	**6.4**	401.8	65.2 (93.3)
	0.1	13.3	17	20.8	99.7	**6.5**	323.4	53.7 (81.4)
	0.3	11	19.7	25.6	99.3	**7.3**	237.6	78.8 (105.5)
	0.5	11.5	23.9	34.4	92.3	**7.9**	263.8	65.8 (97.7)
Flower17	0	8.7	146.72	6.7	36.6	**5.3**	186.5	29.8 (46.7)
	0.1	8.4	163.1	13.8	30.1	**6.2**	138.6	34.8 (53.2)
	0.3	8.3	158.8	31	31.2	**6.4**	111.4	26.9 (42.9)
	0.5	7.5	175.9	28.2	33	**6.8**	118.4	34.4 (52.6)
COIL-20	0	5.9	32.4	5	15.4	**4.4**	131.4	14.6 (20.8)
	0.1	5.2	40.8	10.4	29.2	**5**	77.9	15.1 (22.5)
	0.3	4.6	23.5	10.4	27	**5**	82	13.9 (20.7)
	0.5	6.1	34.2	18.6	27.1	**5.2**	73.6	14.2 (24.9)
ProteinFold	0	4.2	55.7	3.5	24.3	**3.1**	97.6	15 (37.9)
	0.1	4.1	48.6	4.2	20.6	**3.9**	82.3	15.3 (24.3)
	0.3	3.9	66.9	4.8	29.3	**3.9**	67.6	14.6 (23.6)
	0.5	3.6	69.8	6.7	21.8	**4.1**	60.3	12.9 (20.8)
SUN RGB-D	0	818	3804	443	3897	**273**	6445	1682 (2268)
	0.1	704	3891	809	2645	**270**	5529	1560 (2124)
	0.3	520	4241	897	1610	**249**	4128	1295 (1743)
	0.5	461	4048	1151	1615	**235**	3635	1278 (1764)
100leaves	0	22.4	212.7	21.9	58.8	**20.9**	150.6	28.8 (63.1)
	0.1	24.9	156.5	29.6	59.3	**23.4**	196.2	27.1 (43.6)
	0.3	24.4	119.5	38.8	61	**22.3**	167.2	26.2 (41.1)
	0.5	24.4	122.5	30.2	51.1	**21.7**	152	26.9 (64.1)
Caltech-101	0	1097	1123	853	4796	**333**	20238	5718 (5748)
	0.1	833	1492	746	5876	**323**	15335	5792 (7933)
	0.3	470	1481	637	4605	**300**	12002	5215 (7961)
	0.5	250	1469	544	2803	**289**	10421	5222 (7516)

matrices $\left\{\mathbf{C}^{(v)}\right\}_{v=1}^{n_v}$ into a similarity matrix by using Algorithm 2. Then, we perform standard spectral clustering on the similarity matrix to complete IMVC. All similarity matrices produced by DPSC are aggregated for IMVC.

4.1.3 Evaluation Metrics

Following previous work [54], three standard metrics are employed to evaluate the clustering performance of all competing algorithms, including the clustering accuracy (ACC), normalized mutual information (NMI) and F-measure. In the experiments, higher values of these metrics indicate better performance.

4.1.4 Parameter Settings

We assume that the true number of clusters is known for each dataset. The evaluation of the actual number of clusters is beyond the scope of our work. For the changes in the missing ratio during the experiments, we first consider the situation in which all instances available in all the views. Then, we randomly remove a certain percentage of the instances from each view. The missing percentage for each view varies from 10% to 50% in intervals of 20%. In addition, we apply a PCA algorithm to preprocess the existing instances of the samples if the dimensionality of the instances is larger than the number of samples [55]. The existing instances are normalized to $[0, 1]$ in the experiments.

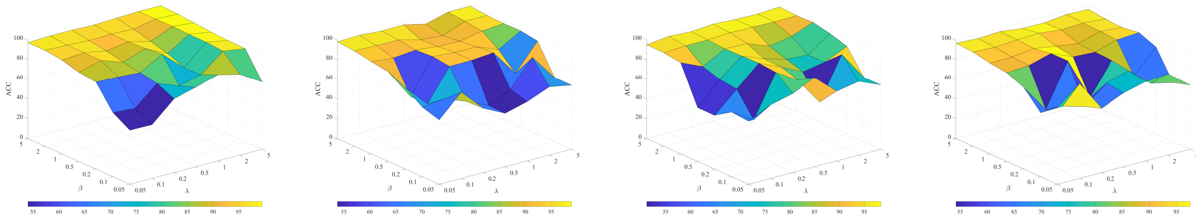
Two parameters λ and β are contained in the proposed LRTL method. To find the best clustering results, the parameter λ is set to $\lambda \in [0.05, 10]$, while the parameter β is tuned in the range of $\{0.05, 0.1, 0.2, 0.5, 1, 2, 5\}$ for the experiments. We employ the grid search approach to find suitable values in the ranges of the parameters λ and β and report the best clustering results. For a fair comparison, we repeat each experiment 10 times and report the average clustering results and the standard deviations for all competing algorithms. For the competing algorithms, we manually tune their parameters to achieve the highest average clustering results. The best and second-best average clustering results are shown in bold and underlined, respectively.

4.2 Clustering Performance Evaluation

Table 3 shows the averages and standard deviations of the ACC (%), NMI (%) and F-measure (%) values for all the competing algorithms with different missing ratios on the nine multiview benchmark datasets. The LRTL algorithm consistently achieves the best clustering results with respect to ACC, NMI and the F-measure in comparison with all the state-of-the art algorithms. For example, the proposed method obtains significant improvements of approximately 19.83%, 17.86%, 20.71% and 16.28% in terms of ACC over the second-best method (TMBDSD) with different missing rates of 0, 10%, 30% and 50% on the Reuters dataset, respectively. Moreover, the proposed LRTL method consistently outperforms all the competing algorithms on the other datasets. This validates the advantages and effectiveness of the proposed LRTL method. In particular, we observe that our method performs slightly better than TMBDSD with improvements of approximately 0.09%, 0.11% and 0.09% in terms of the ACC, NMI and F-measure, respectively, achieved on the Handwritten dataset. This is because these handwritten images of digits lie in a distinct low-dimensional subspace of the ambient space. The low-rank structures of multiple spectral embedding matrices are well preserved by LRTL and TMBDSD. Hence, LRTL and TMBDSD perform dramatically better than the other competing methods. In contrast with the other competing methods, DPSC achieves comparable clustering performance for larger-scale datasets, e.g., the SUN RGB-D and Caltech-101 datasets. This indicates that DPSC exhibits a good ability to learn high-level features from large scale datasets.

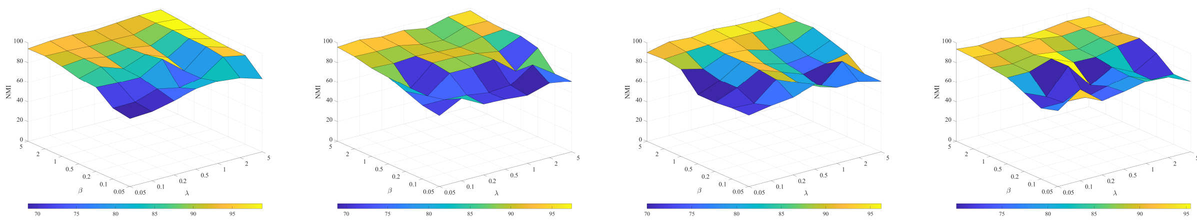
The clustering performance of the proposed method slightly declines as the missing rate increases. Specifically, the differences between the clustering results of the proposed method with missing rates of 0 and 50% are less than 10% on all the datasets. In contrast, the clustering performance of the other competing algorithms is often sensitive to changes in the missing rate. This indicates that the proposed method is more robust than the other approaches. The main reason for this is that the local and global structures of high-dimensional data are effectively explored under the low-rank and sparsity constraints of LRTL. As a result, this alleviates the negative impact of the missing instances.

We also observe that LRTL performs significantly better than $\text{LRTL}_{\textbf{Agg}}$. This shows that the multiview embedding matrix fusion model in LRTL can effectively reveal the essential block structures in high-dimensional data when compared with the summation of the individual adjacency matrices. As expected, the clustering performance worsens



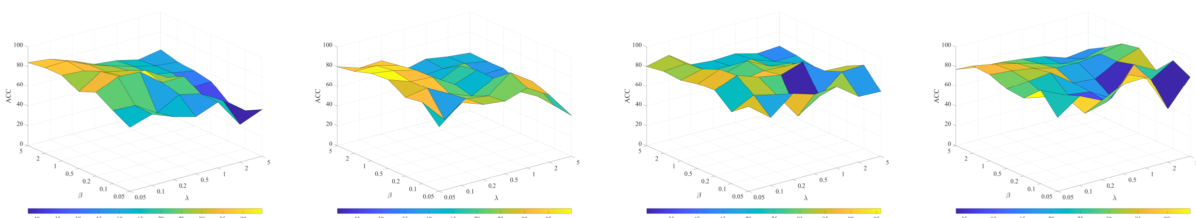
(a) $Ratio = 0$ (b) $Ratio = 0.1$ (c) $Ratio = 0.3$ (d) $Ratio = 0.5$

Fig. 1. The ACC with different λ and β combinations on the Handwritten dataset.



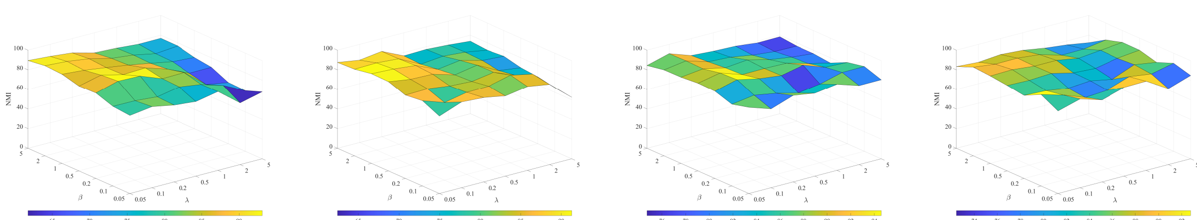
(a) $Ratio = 0$ (b) $Ratio = 0.1$ (c) $Ratio = 0.3$ (d) $Ratio = 0.5$

Fig. 2. The NMI with different λ and β combinations on the Handwritten dataset.



(a) $Ratio = 0$ (b) $Ratio = 0.1$ (c) $Ratio = 0.3$ (d) $Ratio = 0.5$

Fig. 3. The ACC with different λ and β combinations on the COIL-20 dataset.



(a) $Ratio = 0$ (b) $Ratio = 0.1$ (c) $Ratio = 0.3$ (d) $Ratio = 0.5$

Fig. 4. The NMI with different λ and β combinations on the COIL-20 dataset.

in most cases as the missing ratio increases. However, a few slight exceptions are closely related to the random selection of the missing instances in the experiments. In addition, the TMBDSD method achieves the second-best clustering performance on most datasets. However, the clustering performance of TMBDSD often remains unstable on several datasets, e.g., the O-Scene and ProteinFold datasets, as the missing ratio increases. For example, the ACC of the TMBDSD method dramatically drops from 70.8% to 30.13% when the missing ratio increases from 0 to 50%.

Table 4 shows the running times of the algorithms mentioned above on all the datasets with different missing rates (0, 10%, 30% and 50%). The running time of the proposed algorithm usually drops as the missing ratio increases. This is because the number of available instances declines as the missing ratio gradually increases. However, the running time of the proposed algorithm also depends on another important factor, i.e., the number of iterations in Algorithm 3. This is why the running time of the proposed algorithm increases as the missing ratio increases in a few of the experiments. The running times of the proposed algorithm without the alternative fusion strategy are given in the parentheses of Table 4. According to the comparison, the alternative fusion strategy achieves a dramatically reduced computational cost. As theoretically demonstrated, the proposed algorithm dramatically improves the computational efficiency according to the alternative fusion strategy. In

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2022.3230964

addition, it can be seen that the EE-IMVC method executes faster than the other algorithms. However, the clustering results of the EE-IMVC method are worse than those of the proposed method and TMBDSD. Moreover, the proposed method outperforms TMBDSD in terms of running time. Consequently, the running time of the proposed method is average among those of all the competing methods.

4.3 Parameter Sensitivity Analysis

The proposed LRTL method contains two parameters: λ and β. We conduct some experiments to investigate the sensitivity of parameters λ and β in terms of the resulting ACC and NMI. In the experiments, we choose the parameters λ and β from the ranges of $\{0.05, 0.1, 0.2, 0.5, 1, 2, 5\}$ with the grid search strategy. Due to space limitations, two datasets are considered as representatives, i.e., the Handwritten and COIL-20 datasets, for parameter sensitivity analysis with different missing ratios of 0, 10%, 30% and 50%.

Figs. 1-4 show the clustering performance achieved in terms of ACC and NMI with different combinations of the parameters λ and β on the Handwritten and COIL-20 datasets. This indicates that relatively large ranges of the λ and β parameters yield satisfactory clustering results. For example, the clustering results obtained on the two datasets are relatively stable when $\lambda \in [1, 5]$ and $\beta \in [0.05, 0.2]$. In particular, λ is less sensitive than β in the experiments. This is because sparsity is introduced to reduce the impact of the fluctuation of λ to some extent when computing the individual adjacency matrices. It is feasible to employ the grid search approach to find suitable parameters if prior knowledge of the datasets is available in practice.

4.4 Diagonal Block Structure Analysis

The diagonal block structures play critical roles in the proposed method. In Problem (27), $\mathbf{H}^{(v)}\left(\mathbf{H}^{(v)}\right)^T$ is a low-rank matrix containing k diagonal block submatrices under ideal conditions. Moreover, the diagonal block structure property of $\mathbf{H}^{(v)}\left(\mathbf{H}^{(v)}\right)^T$ is closely related to $\mathbf{W}^{(v)}$ in Equation (23). Specifically, $\mathbf{H}^{(v)}\left(\mathbf{H}^{(v)}\right)^T$ must contain diagonal block submatrices if $\mathbf{W}^{(v)}$ is a strict diagonal block structure, but the converse is not always true. In the experiments, each dataset contains 3 or more views. For simplicity, we conduct an experiment to investigate the block structures of the special matrices produced by the summation of $\left\{\mathbf{W}^{(v)}\right\}_{v=1}^{n_v}$ with different missing ratios. The block structure of an individual $\mathbf{W}^{(v)}$ is much stricter than that of the corresponding special matrix. The parameters of the proposed method are set according to Section 4.2.

Fig. 5 shows four examples of the block structures of the similarity matrices produced with different missing ratios on the Reuters dataset. For example, six distinct diagonal block submatrices are located along the diagonal direction of the special matrix in Fig. 5a. The number of diagonal block submatrices is equal to the number of clusters in the Reuters dataset. In addition, six diagonal block submatrices are accurately observed with some noise in Figs. 5b-5c as the missing rate gradually increases from 10% to 30%. However, the diagonal block submatrices become insufficiently clear in Fig. 5d since the missing rate adds up to 50%. The experimental results validate the block structures of $\mathbf{H}^{(v)}\mathbf{H}^{(v)T}$ in Problem (27).

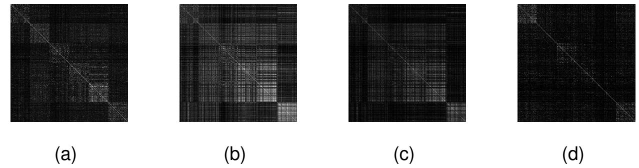


Fig. 5. Examples of the block structures of the similarity matrices produced with different missing ratios on the Reuters dataset. (a) $Ratio = 0$. (b) $Ratio = 0.1$. (c) $Ratio = 0.3$. (d) $Ratio = 0.5$.

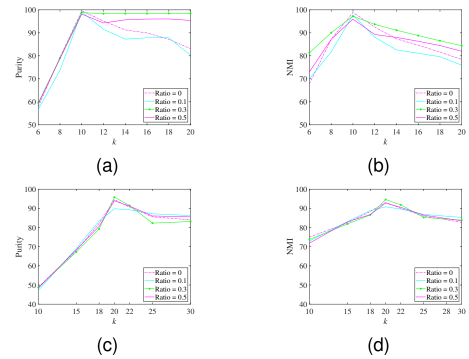


Fig. 6. The purity and NMI values obtained with respect to different number of clusters on the Handwritten and COIL-20 datasets. (a) The purity achieved on Handwritten. (b) The NMI attained on Handwritten. (c) The purity achieved on COIL-20. (d) The NMI attained on COIL-20.

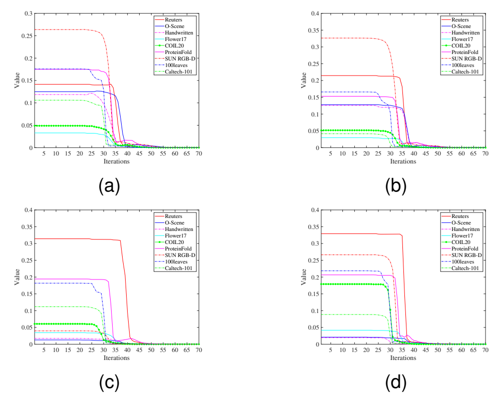


Fig. 7. Convergence results of Algorithm 3 with different missing ratios on all the datasets. (a) $Ratio = 0$. (b) $Ratio = 0.1$. (c) $Ratio = 0.3$. (d) $Ratio = 0.5$.

4.5 Empirical Study on the Number of Clusters

The number of clusters k is assumed to be known in the above experiments. However, the number of clusters may be unknown in practice. Hence, we investigate the effect induced by varying the number of clusters involved in the proposed LRTL method. Two metrics are employed to measure the clustering quality achieved under different numbers of clusters, i.e., the clustering purity and NMI [54]. We set the parameters of the proposed method according to Section 4.2.

Fig. 6 shows the clustering performance achieved in terms of the purity and NMI values obtained with different number of clusters on the Handwritten and COIL-20 datasets. It is observed that an increase in the number of clusters results in higher purity and NMI values until the true number of clusters for these two datasets is reached. Then the purity and NMI values decline slowly as the number of clusters continues to steadily increase. This indicates that the relatively large number of clusters still produces encouraging clustering results for the proposed method. Therefore, we empirically suggest setting a relatively large number of clusters if no prior knowledge concerning the number of clusters is available.

4.6 Convergence Analysis

We analyze the convergence property of the proposed method on the nine datasets. Fig. 7 shows the convergence curves of Algorithm 3 with different missing ratios (0, 10%, 30% and 50%) on all the datasets. For each figure, the x-axis represents the number of iterations, and the y-axis denotes the absolute value of the convergence condition $\|\mathcal{G}_t - \mathcal{T}_t\|_{\max}$ for Algorithm 3. From the figures, we observe that the value of the convergence condition remains steady within approximately 30-40 iterations and then starts to monotonically decrease, quickly converging to a steady state. It usually converges in less than 70 iterations in the experiments. These results demonstrate the good convergence of the proposed method in practice, although it is difficult to prove the convergence of Algorithm 3 in theory.

5 Conclusion

In this paper, we propose an LRTL method that successfully learns a consensus low-dimensional embedding matrix for IMVC. Individual low-dimensional embedding matrices are learned from incomplete multiview data via the self-expressiveness property of high-dimensional data. Compared with the intuitive combination of low-rank and sparsity regularizations, the global and local structures of multiview data are explicitly captured by considering successive low-rank and sparsity constraints. In addition, we present a multiview embedding matrix fusion model to achieve a consensus low-dimensional embedding matrix for the k-means algorithm. This model effectively exploits complementary information by finding the high-order correlations of multiple views. To improve the computational efficiency of our approach, we present an alternative fusion strategy for the fusion model by considering the relations among multiple views. Finally, extensive experimental results obtained on nine datasets demonstrate the superiority of the proposed LRTL method over other state-of-the-art approaches.

References

[1] R. Wang, W. Ji, M. Liu, X. Wang, J. Weng, S. Denge, S. Gao, and C. Yuang, "Review on mining data from multiple data sources," *Pattern Recognit. Lett.*, vol. 109, pp. 120–128, Jul. 2018.

[2] Z. Deng, R. Liu, P. Xu, K. Choi, W. Zhang, X. Tian, T. Zhang, L. Liang, B. Qin, and S. Wang, "Multi-view clustering with the cooperation of visible and hidden views," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 2, pp. 803–815, Feb. 2022.

[3] H. Xiao, Y. Chen, and X. Shi, "Knowledge graph embedding based on multi-view clustering framework," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 2, pp. 585–596, Feb. 2021.

[4] Y. Li, M. Yang, D. Peng, T. Li, J. Huang, and X. Peng, "Twin contrastive learning for online clustering," *Int. J. Comput. Vis.*, vol. 130, p. 2205–2221, Jul. 2022.

[5] M. Yang, Y. Li, P. Hu, J. Bai, J. Lv, and X. Peng, "Robust multi-view clustering with incomplete information," *IEEE Trans. Pattern Anal. and Mach. Intell.*, pp. 1–14, Mar. 2022.

[6] X. Peng, Y. Li, I. W. Tsang, J. L. H. Zhu, and J. T. Zhou, "Xai beyond classification: Interpretable neural clustering," *J. Mach. Learn. Res.*, vol. 23, no. 6, pp. 1–28, Jul. 2022.

[7] Y. Xie, J. Liu, Y. Qu, D. Tao, W. Zhang, L. Dai, and L. Ma, "Robust kernelized multiview self-representation for subspace clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 2, pp. 868–881, Feb. 2021.

[8] C. Xu, Z. Guan, W. Zhao, Q. Wu, M. Yan, L. Chen, and Q. Miao, "Recommendation by users' multimodal preferences for smart city applications," *IEEE Trans. Industr. Inform.*, vol. 17, no. 6, pp. 4197–4205, Jun. 2021.

[9] Y. Tang, Y. Xie, X. Yang, J. Niu, and W. Zhang, "Tensor multi-elastic kernel self-paced learning for time series clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 3, pp. 1223–1237, Mar. 2021.

[10] J. Chen, S. Yang, H. Mao, and C. Fahy, "Multiview subspace clustering using low-rank representation," *IEEE Trans. Cybern.*, pp. 1–15, Jun. 2021.

[11] C. Zhang, H. Fu, J. Wang, W. Li, X. Cao, and Q. Hu, "Tensorized multi-view subspace representation learning," *Int. J. Comput. Vis.*, vol. 128, no. 8, pp. 2344–2361, Aug. 2020.

[12] X. Zhu, S. Zhang, W. He, R. Hu, C. Lei, and P. Zhu, "One-step multi-view spectral clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 10, pp. 2022–2034, Oct. 2019.

[13] Z. Guan, L. Zhang, J. Peng, and J. Fan, "Multi-view concept learning for data representation," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 11, pp. 3016–3028, Nov. 2015.

[14] Y. Xie, B. Lin, Y. Qu, C. Li, W. Zhang, L. Ma, Y. Wen, and D. Tao, "Joint deep multi-view learning for image clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 11, pp. 3594–3606, Nov. 2021.

[15] L. Li, Z. Wan, and H. He, "Incomplete multi-view clustering with joint partition and graph learning," *IEEE Trans. Knowl. Data Eng.*, pp. 1–15, May 2021.

[16] X. Liu, M. Li, C. Tang, J. Xia, J. Xiong, L. Liu, M. Kloft, and E. Zhu, "Efficient and effective regularized incomplete multi-view clustering," *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. 43, no. 8, pp. 2634–2646, Aug. 2021.

[17] Z. Li, C. Tang, X. Liu, X. Zheng, W. Zhang, and E. Zhu, "Tensor-based multi-view block-diagonal structure diffusion for clustering incomplete multi-view data," in *IEEE Int. Conf. on Multimedia Expo*, Online, Jul. 2021, pp. 1–6.

[18] C. Xu, H. Liu, Z. Guan, X. Wu, J. Tan, and B. Ling, "Adversarial incomplete multiview subspace clustering networks," *IEEE Trans. Cybern.*, vol. 52, no. 10, pp. 10 490–10 503, Oct. 2022.

[19] Z. Tao, J. Li, H. Fu, Y. Kong, and Y. Fu, "From ensemble clustering to subspace clustering: Cluster structure encoding," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–12, Sept. 2021.

[20] J. Wen, Z. Zhang, Z. Zhang, L. Zhu, L. Fei, B. Zhang, and Y. Xu, "Unified tensor framework for incomplete multi-view clustering and missing-view inferring," in *in Proc. AAAI Conf. Artif. Intell.*, May 2021, pp. 10 273–10 281.

[21] 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," *Int. J. Comput. Vis.*, vol. 126, no. 11, pp. 1157–1179, Apr. 2018.

[22] Z. Wei, C. Xu, Z. Guan, and Y. Liu, "Multiview concept learning via deep matrix factorization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 2, pp. 814–825, Feb. 2021.

[23] M. Hu and S. Chen, "One-pass incomplete multi-view clustering," in *in Proc. AAAI Conf. Artif. Intell.*, Jan. 2019, pp. 3838–3845.

[24] X. Liu, X. Zhu, M. Li, L. Wang, C. Tang, J. Yin, D. Shen, H. Wang, and W. Gao, "Late fusion incomplete multi-view clustering," *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. 41, no. 10, pp. 2410–2423, Oct. 2018.

[25] Q. Wang, Z. Tao, W. Xia, Q. Gao, X. Cao, and L. Jiao, "Adversarial multiview clustering networks with adaptive fusion," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–13, Feb. 2022.

[26] C. Zhang, Y. Cui, Z. Han, J. T. Zhou, H. Fu, and Q. Hu, "Deep partial multi-view learning," *IEEE Trans. Pattern Anal. and Mach. Intell.*, pp. 1–14, Nov. 2020.

[27] Y. Wang, W. Huang, F. Sun, T. Xu, Y. Rong, and J. Huang, "Deep multimodal fusion by channel exchanging," in *in Proc. 34th Adv.*

Neural. Inf. Process. Syst., Vancouver, Canada, Dec. 2020, pp. 4835–4845.

[28] L. Zhou, X. Bai, D. Wang, X. Liu, J. Zhou, and E. Hancock, "Latent distribution preserving deep subspace clustering," in *in Proc. 28th Int. Joint Conf. Artif. Intell.*, Macao, China, Aug. 2019, pp. 4440–4446.

[29] C. Xu, Z. Guan, W. Zhao, Y. Niu, Q. Wang, and Z. Wang, "Deep multi-view concept learning," in *in Proc. 27th Int. Joint Conf. Artif. Intell.*, Stockholm, Sweden, Jul. 2018, pp. 2898–2904.

[30] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. S., and A. Y. Wu, "An efficient k-means clustering algorithm: analysis and implementation," *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. 24, no. 7, pp. 881–892, Jul. 2002.

[31] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan, "Tensor robust principal component analysis with a new tensor nuclear norm," *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. 42, no. 4, pp. 925–938, Apr. 2020.

[32] M. E. Kilmer and C. D. Martin, "Factorization strategies for third-order tensors," *Linear Algebra Its Appl.*, vol. 435, no. 3, pp. 641–658, Aug. 2011.

[33] Y. Xie, W. Zhang, Y. Qu, L. Dai, and D. Tao, "Hyper-laplacian regularized multilinear multiview self-representations for clustering and semisupervised learning," *IEEE Trans. Cybern.*, vol. 50, no. 2, pp. 572–586, Feb. 2020.

[34] J. Shi, J. Malik, and S.Sastry, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. 22, no. 8, pp. 181–214, Aug. 2000.

[35] M. E. Kilmer, K. Braman, N. Hao, and R. C. Hoover, "Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging," *SIAM J. Matrix Anal. Appl.*, vol. 34, no. 1, pp. 148–172, Aug. 2013.

[36] P. Favaro and R. Vidal, "A closed form solution to robust subspace estimation and clustering," in *in Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Colorado Springs, CO, USA, Jun. 2011, pp. 1801–1807.

[37] J. Chen, H. Mao, Y. Sang, and Z. Yi, "Subspace clustering using a symmetric low-rank representation," *Knowledge-Based Systems*, vol. 127, no. 1, pp. 46–57, Jul. 2017.

[38] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, "Efficient projections onto the l_1-ball for learning in high dimensions," in *Proc. 25th Int. Conf. Mach. Learn. (ICML)*, Helsinki, Finland, Jul. 2008, pp. 272–279.

[39] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Mar. 2011.

[40] J. Huang, F. Nie, and H. Huang, "Spectral rotation versus k-means in spectral clustering," in *in Proc. AAAI Conf. Artif. Intell.*, Bellevue, Washington, USA, Jul. 2013, pp. 431–437.

[41] B. Gao, X. Liu, X. Chen, and Y. Yuan, "A new first-order algorithmic framework for optimization problems with orthogonality constraints," *SIAM J. Optim.*, vol. 28, no. 1, pp. 302–332, Feb. 2018.

[42] Z. Zhang, L. Liu, F. Shen, H. Shen, and L. Shao, "Binary multi-view clustering," *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. 41, no. 7, pp. 1774–1782, Jul. 2019.

[43] M. Thoma. (2017, july) The reuters dataset. [Online]. Available: https://martin-thoma.com/nlp-reuters

[44] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comput. Vis.*, vol. 42, no. 3, p. 145–175, May 2001.

[45] A. Asuncion and D. Newman. (2007) UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences. [Online]. Available: http://archive.ics.uci.edu/ml/

[46] M. E. Nilsback and A. Zisserman, "A visual vocabulary for flower classification," in *in Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, New York, USA, june 2006, pp. 1447–1454.

[47] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia object image library (coil-20)," Tech. Rep. Technical Report CUCS-005-96, Feb. 1996.

[48] S. Song, S. P. Lichtenberg, and J. Xiao, "Sun rgb-d: a rgb-d scene understanding benchmark suite," in *in Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, Jun. 2015, pp. 567–576.

[49] J. Liu, X. Liu, Y. Yang, L. Liu, S. Wang, W. Liang, and J. Shi, "One-pass multi-view clustering for large-scale data," in *in Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oline, Oct. 2021, pp. 12 344–12 353.

[50] C. Mallah, J. Cope, and J. Orwell, "Plant leaf classification using probabilistic integration of shape, texture and margin features," *Signal Process. Pattern Recognit. Appl.*, vol. 5, no. 1, p. 45–54, 2013.

[51] D. Dua and C. Graff. (2017) UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences. [Online]. Available: http://archive.ics.uci.edu/ml

[52] J. Liu, X. Liu, Y. Zhang, P. Zhang, W.Tu, S. Wang, S. Zhou, W. Liang, S. Wang, and Y. Yang, "Self-representation subspace clustering for incomplete multi-view data," in *ACM Int. Conf. Multimedia*, Chengdu, China, Oct. 2021, pp. 1–11.

[53] J. Wen, Z. Zhang, Z. Zhang, L. Fei, and M. Wang, "Generalized incomplete multi-view clustering with flexible locality structure diffusion," *IEEE Trans. Cybern.*, vol. 51, no. 1, pp. 101–114, Jan. 2021.

[54] J. Chen, H. Mao, Z. Wang, and X. Zhang, "Low-rank representation with adaptive dictionary learning for subspace clustering," *Knowl. Based Syst.*, vol. 223, pp. 1–12, July 2021.

[55] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *in Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Lahaina, Maui, Hawaii, USA, June 1991, pp. 586–587.

Jie Chen received the BSc degree in Software Engineering, MSc degree and PhD degree in Computer Science from Sichuan University, Chengdu, China, in 2005, 2008 and 2014, respectively. From 2008 to 2009, he was with Huawei Technologies Co., Ltd. as a software engineer. He is currently an Associate Professor with the College of Computer Science, Sichuan University, China. His current research interests include machine learning, big data analysis, and deep neural networks.

Zhu Wang received the B.M., MSc., and LLD. degrees in Civil and Commercial Law from Renmin University of China, China in 2003, 2006, and 2009, respectively. He is currently a Professor in Law and Director of Institute of Rule of Law of Market Economy, Sichuan University, Chengdu, China. His research interests are mainly in Tort, insurance law, constitution and big data analysis of law.

Hua Mao received the B.S. degree and M.S. degree in Computer Science from University of Electronic Science and Technology of China (UESTC) in 2006 and 2009, respectively. She received her Ph.D. degree in Computer Science and Engineering from Aalborg University, Denmark in 2013. She is currently a Senior Lecturer in Department of Computer and Information Sciences, Northumbria University, U.K. Her current research interests include Deep Neural Networks and Big Data.

Xi Peng (Member, IEEE) is currently a Full Professor with the College of Computer Science, Sichuan University. His current research interest includes machine intelligence and has authored more than 50 articles in these areas. He has served as an Associate Editor/Guest Editor for six journals, including the *IEEE Transactions on SMC: Systems* and *IEEE Transactions on Neural Network And Learning Systems* and the Area Chair/Senior Program Committee Member for the conferences such as IJCAI, AAAI, and ICME.