Jointly learning kernel representation tensor and affinity matrix for multi-view clustering

Yongyong Chen, Xiaolin Xiao and Yicong Zhou, *Senior Member, IEEE,*

***Abstract*—Multi-view clustering refers to the task of partitioning numerous unlabeled multimedia data into several distinct clusters using multiple features. In this paper, we propose a novel nonlinear method called joint learning multi-view clustering (JLMVC) to jointly learn kernel representation tensor and affinity matrix. The proposed JLMVC has three advantages: (1) unlike existing low-rank representation-based multi-view clustering methods that learn the representation tensor and affinity matrix in two separate steps, JLMVC jointly learns them both. (2) using the "kernel trick", JLMVC can handle nonlinear data structures for various real applications. (3) different from most existing methods that treat representations of all views equally, JLMVC automatically learns a reasonable weight for each view. Based on the alternating direction method of multipliers, an effective algorithm is designed to solve the proposed model. Extensive experiments on eight multimedia datasets demonstrate the superiority of the proposed JLMVC over state-of-the-art methods.**

***Index Terms*—Multi-view clustering, low-rank tensor representation, kernel trick, affinity matrix, adaptive weight**

I. Introduction

In many real-world applications, multimedia data such as images, videos, audio, and documents, are usually represented by different features or collected from various fields (called *multi-view data*) [1–3]. For example, in multimedia retrieval [2], images can be represented by color, textures, and edges. In video surveillance [3], the same scene is monitored by multiple cameras from different viewpoints. In natural language processing [4], documents can be translated by multiple different languages like Chinese, English, French, and so on. Considering that multi-view data are greatly conducive to the performance improvement, multi-view clustering has attracted great research interests in many fields including multimedia data mining, machine learning and pattern recognition communities [5–8].

Given multi-view features extracted from the original multimedia data, they are used to partition all unlabeled multimedia data into several distinct clusters. Massive approaches for clustering have been proposed. Either single-view clustering or multi-view clustering, they usually follow two main steps: 1) constructing a symmetric affinity matrix (also called similarity matrix) to describe the pairwise relations between multimedia data points and 2) performing the spectral clustering algorithm [9] to obtain clustering results. The core of these methods is construction of the affinity matrix. This means that the quality of the learned affinity matrix heavily determines the clustering performance. In literature, two common schemes, the raw multimedia features and computed representations [10, 11], are selected to conduct the affinity matrix, leading to the following three categories: 1) graph-based methods [12–18], 2) subspace clustering-based methods [5–8, 11, 19–23], 3) their combinations [10, 24, 25]. For example, due to simplicity and effectiveness, k-Nearest Neighbor using cosine or heat kernel distances [26] has become an intuitive way to construct the affinity matrix. Following the idea that local connectivity of multimedia data can be measured by the Euclidean distance, the work in [12] constructed the affinity matrix by assigning adaptive neighbors to each multimedia data point. In [13], Nie *et al.* adopted the l_1-norm distance instead of the Euclidean distance and proposed a graph clustering relaxation. Based on the fact that the affinity matrix should obey the block diagonal property, Nie *et al.* [14] imposed the rank constraint on the Laplacian matrix for graph-based clustering. To well explore the complementary information of multi-view features, the approaches in [16] and [17] extended the adaptive neighbor strategy [12] and the rank constraint [14] from the single-view setting into the multi-view one, respectively. Following this, Wang *et al*. [18] pursued a unified affinity matrix from the affinity matrices of all views and the rank function was considered to partition multimedia data points into optimal number of clusters. However, these graph-based approaches, *e.g.*, [16–18], usually construct the affinity matrix by directly using the raw multimedia features which are often corrupted by noise and outliers. Thus, they may obtain an unreliable and inaccurate affinity matrix [10, 25].

As the second category, subspace clustering-based methods have become the mainstream due to their excellent interpretability and performance. The goal of subspace clustering is to simultaneously find low-dimensional subspaces and partition multimedia data points into multiple subspaces. Specifically, sparse subspace clustering (SSC) [20] and low-rank representation (LRR) [19] are two representative works, resulting in a local representation matrix and a global one, respectively. Since SSC learns the representation matrix by l_1-norm, it imposes the sparsity on all entries of the representation matrix. However, LRR conducts the representation matrix by the low-rank regularizer. This imposes the sparsity on the singular values. Beyond the low-rankness and sparsity, some extra

This work was funded in part by The Science and Technology Development Fund, Macau SAR (File no. 189/2017/A3), and by the Research Committee at University of Macau under Grants MYRG2016-00123-FST and MYRG2018-00136-FST. (Corresponding author: Yicong Zhou.)

Y. Chen and Y. Zhou are with the Department of Computer and Information Science, University of Macau, Macau 999078, China (e-mails:YongyongChen.cn@gmail.com and yicongzhou@um.edu.mo).

X. Xiao is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China and also with the Department of Computer and Information Science, University of Macau, Macau 999078, China (email: shellyxiaolin@gmail.com).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2019.2952984, IEEE Transactions on Multimedia

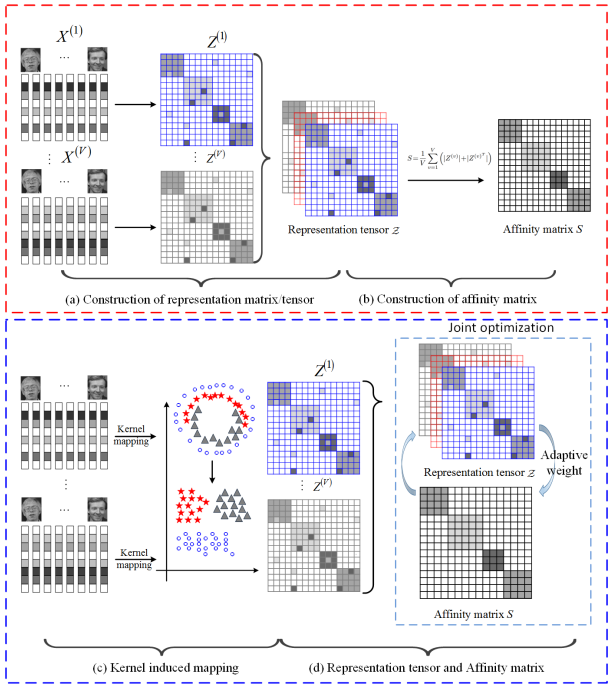

Fig. 1. Comparison of existing low-rank tensor representation-based MVC methods (the red dashed rectangle) and our proposed JLMVC (the blue dashed rectangle). Existing methods construct the representation matrix (a) and the affinity matrix (b) in two separate steps without considering their correlation. JLMVC learns the representation tensor and the affinity matrix (d) in a unified framework. Additionally, the kernel-induced mapping is adopted to map the original multimedia data (usually nonlinear separable) into a new linear space.

structures underlying data, such as the local similarity structure and nonnegativity, may not be fully considered. Instead of the fixed dictionary, *i.e.*, the original multimedia feature, the work in [27] proposed to learn a locality-preserving dictionary to capture the intrinsic geometric structure of the dictionary for LRR. Yin *et al*. [25] proposed to integrate LRR and the graph construction in a unified framework to learn an adaptive low-rank graph affinity matrix. A similar idea was adopted in [10, 24]. A major challenge is that, when handing multi-view features, they may cause a significant performance degradation since they focus only on single-view feature.

Recently, considerable efforts based on deep neural network have been expended for clustering. For example, Ji *et al*. [28] proposed a deep neural network by introducing a self-expressive layer into the auto-encoder framework for clustering. To conduct a deep structure, the authors in [29] adopted semi-nonnegative matrix factorization for mutli-view clustering. In [30], a highly-economized scalable image clustering method was proposed to cluster large-scale multi-view images. Besides, to deal with multi-view clustering with missing features, Chao *et al*. [31] presented an enhanced multi-view co-clustering method. For a comprehensive survey on clustering, please refer to [32] and the references therein.

A. Related work

The existing low-rank-based approaches for multi-view clustering can be roughly grouped into two categories: two-dimension matrix-based low-rank methods [5, 22, 33–38] and three-dimension tensor-based low-rank ones [6–8]. For example, to deal with multiple multimedia features, the work in [33] proposed to concatenate all heterogeneous features and then perform LRR [19]. Xia *et al*. [34] exploited the low-rank and sparse matrix decomposition to uncover a shared transition probability matrix under the Markov chain method. Except for consistency among multi-view features, the work in [36] took local view-specific information into consideration for multi-view clustering. Similarly, Tang *et al*. [5] proposed a multi-view clustering method by learning a joint affinity graph. In [5, 36], the consistency measures the common properties among all views while the specificity captures the inherent difference in each view. Different from these approaches that use the nuclear norm to depict the low-rank property of the representation matrices, Wang *et al*. [22] proposed to factorize each representation matrix as the product of symmetric low-rank data-cluster matrices, such that the singular value decomposition can be ignored. Following this, Liu *et al*. [38] proposed to mine a consensus representation of all views by multi-view non-negative matrix factorization.

The most representative methods of the second category are the tensor unfolding-based method (LT-MSC) [6] and t-singular value decomposition (t-SVD)-based one (t-SVD-MSC) [7]. As shown in Fig. 1 (a), each representation matrix is stored as the frontal slice of a tensor, resulting in a third-order tensor (called *representation tensor*). The main difference between [6] and [7] is the tensor rank approximation which aims to explore the high order correlations among multi-views. By organizing all multi-view features into a third-order tensor, the work in [39] exploited the sparsity and tensor nuclear norm penalty with self-expressiveness to construct the representation tensor.

Although these approaches have achieved a great advance for multi-view clustering, they may suffer from the following challenges: 1) their performance may sharply degrade in real applications when the multimedia data come from nonlinear subspaces. The intuitive reason is that they were originally designed to deal with the data that lie within multiple linear subspaces [8, 40, 41]. 2) the correlation between the representation tensor and affinity matrix may not be fully exploited [42]. They learn the representation tensor via different low-rank tensor representations, and then construct the affinity matrix as shown in Figs. 1 (a) and (b) in two separate steps. This means that the global optimal affinity matrix cannot be ensured. 3) the importance of each view in the construction of the affinity matrix is not considered. For example, methods in [6, 7, 43] simply average all representation matrices with the same weight. The approach in [43] overcomes the first limitation, but fails to address the other two challenges. To our best knowledge, no work has been done to address these three challenges simultaneously.

B. Our contributions

To address above three challenges, we propose a unified model to jointly learn the kernel representation tensor and affinity matrix for multi-view clustering (JLMVC). JLMVC learns the representation tensor and affinity matrix jointly

such that their correlations can be well exploited, handles the nonlinear multimedia data using a kernel-induced mapping, and adopts the adaptive weight strategy to form a unified affinity matrix. Fig. 1 compares the proposed JLMVC with two state-of-the-art low-rank tensor representation-based MVC methods LT-MSC [6] and t-SVD-MSC [7]. As can be observed that, under the assumption that the original data lie within multiple linear subspaces, existing low-rank tensor representation-based MVC methods learn the representation tensor from the original multimedia data. However, this assumption may not be ensured in real applications. To achieve nonlinear multi-view clustering, JLMVC maps the original multimedia data from the input data space into a new feature space such that the mapped data points can reside in multiple linear subspaces, as shown in the middle of Fig. 1 (c). JLMVC then learns the representation tensor and affinity matrix simultaneously. Finally, the learned unified affinity matrix is fed to the input of the spectral clustering algorithm [9] to obtain the clustering results.

The contributions and novelty of this paper are summarized as follows:

- We propose a joint learning multi-view clustering (JLMVC) model to jointly learn kernel representation tensor and affinity matrix for multi-view clustering. JLMVC is able to well explore the correlation between the representation tensor and affinity matrix, handles the nonlinear data using a kernel-induced mapping, and adopts the adaptive weight strategy to form a unified affinity matrix.
- JLMVC uses the tensor nuclear norm to encode the low rank property of the representation tensor and adaptively learns different weights for different views' representation matrices. This greatly benefits the construction of the unified affinity matrix.
- An effective algorithm is designed to solve the JLMVC model via the alternating direction method of multipliers. Extensive experiments on eight popular multimedia datasets are conducted and validate the superiority of JLMVC over ten state-of-the-art approaches.

C. Organization of the paper

The rest of this paper is structured as follows. Section II introduces some notations and preliminaries, especially the t-SVD-based tensor nuclear norm which is used to depict the low-rank property of the representation tensor. In Section III, we introduce JLMVC and design an iterative algorithm under the alternating direction method of multipliers framework. We evaluate the performance of the proposed JLMVC on eight real-world multi-view datasets in Section IV and conclude the whole paper in Section V.

II. Notations and Preliminaries

In this section, we aim to introduce some notations used throughout this paper and the t-SVD-based tensor nuclear norm (see *Definition* 2.2) that will be used to depict the low-rank property of the representation tensor. Some basic notations are summarized in Table I.

TABLE I
Basic notations and their descriptions.

Notation	Meaning
$\mathcal{X}$, X, x	tensor, matrix, vector
$\mathcal{X}^{(k)}$	the k-th frontal slice of tensor $\mathcal{X}$
$\hat{\mathcal{X}} = \textbf{fft}(\mathcal{X}, [], 3)$	fast Fourier transformation along tube fiber
n, V	the number of samples, views
d_v	feature dimension of the v-th view
$X^{(v)} \in \mathbb{R}^{d_v \times n}$	feature matrix of the v-th view
$\mathcal{Z} \in \mathbb{R}^{n \times n \times V}$	the representation tensor
$S \in \mathbb{R}^{n \times n}$	the affinity matrix
$E^{(v)} \in \mathbb{R}^{d_v \times n}$	the sample-specific corruptions
$\|\cdot\|_{2,1}$, $\|\cdot\|_F$	$l_{2,1}$-norm, Frobenius norm
$\|\cdot\|_{\circledast}$, $\|\cdot\|_\infty$	t-SVD-nuclear norm, infinity norm
$\mathbb{R}$, $\mathcal{H}$	the real space, the kernel Hilbert space
$\mathbf{K}^{(v)} \in \mathbb{R}^{n \times n}$	the kernel matrix

Before the definition of t-SVD [44], several operators are first introduced. For a tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, its block circular matrix $\textbf{bcirc}(\mathcal{X})$ and block diagonal matrix $\textbf{bdiag}(\mathcal{X})$ are defined as

$$\textbf{bcirc}(\mathcal{X}) = \begin{bmatrix} \mathcal{X}^{(1)} & \mathcal{X}^{(n_3)} & \cdots & \mathcal{X}^{(2)} \\ \mathcal{X}^{(2)} & \mathcal{X}^{(1)} & \cdots & \mathcal{X}^{(3)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{X}^{(n_3)} & \mathcal{X}^{(n_3-1)} & \cdots & \mathcal{X}^{(1)} \end{bmatrix},$$

$$\textbf{bdiag}(\mathcal{X}) = \begin{bmatrix} \mathcal{X}^{(1)} & & & \\ & \mathcal{X}^{(2)} & & \\ & & \ddots & \\ & & & \mathcal{X}^{(n_3)} \end{bmatrix}.$$

The block vectorization is defined as $\textbf{bvec}(\mathcal{X}) = [\mathcal{X}^{(1)}; \cdots; \mathcal{X}^{(n_3)}]$. The inverse operations of **bvec** and **bdiag** are defined as $\textbf{bvfold}(\textbf{bvec}(\mathcal{X})) = \mathcal{X}$ and $\textbf{bdfold}(\textbf{bdiag}(\mathcal{X})) = \mathcal{X}$, respectively. Let $\mathcal{Y} \in \mathbb{R}^{n_2 \times n_4 \times n_3}$. The **t-product** $\mathcal{X} * \mathcal{Y}$ is an $n_1 \times n_4 \times n_3$ tensor, $\mathcal{X} * \mathcal{Y} = \textbf{bvfold}(\textbf{bcirc}(\mathcal{X}) * \textbf{bvec}(\mathcal{Y}))$. The **transpose** of $\mathcal{X}$ is $\mathcal{X}^T \in \mathbb{R}^{n_2 \times n_1 \times n_3}$ by transposing each of the frontal slices and then reversing the order of transposed frontal slices 2 through n_3. The **identity tensor** $\mathcal{I} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$ is a tensor whose first frontal slice is an $n_1 \times n_1$ identity matrix and the rest frontal slices are zero. A tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$ is **orthogonal** if it satisfies $\mathcal{X}^T * \mathcal{X} = \mathcal{X} * \mathcal{X}^T = \mathcal{I}$.

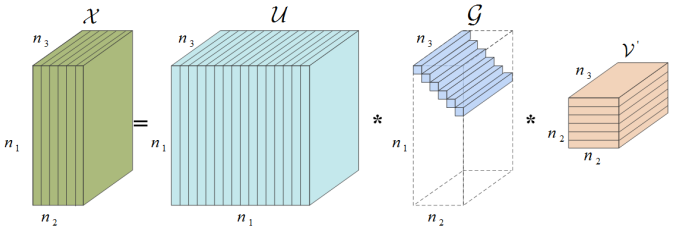

Fig. 2. The t-SVD of a tensor of size $n_1 \times n_2 \times n_3$.

Definition 2.1: **(t-SVD)** Given $\mathcal{X}$, its t-SVD is defined as

$$\mathcal{X} = \mathcal{U} * \mathcal{G} * \mathcal{V}^T,$$

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, $\mathcal{G} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is an f-diagonal tensor. Each of its frontal slices is a diagonal matrix.

Fig. 2 shows the t-SVD of a third-order tensor. The t-SVD-based tensor nuclear norm (t-SVD-TNN) is given as follows.

Definition 2.2: **(t-SVD-TNN)** The t-SVD-TNN of a tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, denoted as $\|\mathcal{X}\|_{\circledast}$, is defined as the sum of singular values of all the frontal slices of $\hat{\mathcal{X}}$, *i.e.*,

$$\|\mathcal{X}\|_{\circledast} = \sum_{i=1}^{\min\{n_1,n_2\}} \sum_{k=1}^{n_3} |\hat{\mathcal{G}}(i,i,k)|. \tag{1}$$

III. Joint learning multi-view clustering

In this section, we first elaborate the proposed JLMVC model in Section III-A, and then solve this model by the alternating direction method of multipliers (ADMM) in Section III-B. Considering that, in real world applications, the multimedia data may be drawn from multiple nonlinear subspaces, JLMVC first uses the kernel trick to solve the nonlinearity. Based on the self-expression property [19, 20], JLMVC carries out joint learning of the representation tensor and unified affinity matrix.

A. Problem formulation

The existing multi-view clustering method t-SVD-MSC [7] learns the representation tensor $\mathcal{Z}$ by

$$\begin{aligned} &\min_{\mathcal{Z},E} \|\mathcal{Z}\|_{\circledast} + \alpha \sum_{v=1}^{V} \|E^{(v)}\|_{2,1} \\ &s.t. \quad X^{(v)} = X^{(v)} Z^{(v)} + E^{(v)}, \quad v = 1, \cdots, V, \\ &\qquad \mathcal{Z} = \Phi(Z^{(1)}, Z^{(2)}, \cdots, Z^{(V)}). \end{aligned} \tag{2}$$

where $X^{(v)} \in \mathbb{R}^{d_v \times n}$ denotes the v-th view feature; $\alpha > 0$ is the regularization parameter; E denotes noise and outliers; $\Phi(\cdot)$ is an operator to stack all representation matrices $\{Z^{(v)}\}$ into a third-order tensor $\mathcal{Z}$ as shown in Fig. 1 (a).

Once $\mathcal{Z}$ is yielded by Eq. (2), the affinity matrix S is constructed by averaging all frontal slices of $\mathcal{Z}$. This means that, in the construction of S, the correlation between S and $\mathcal{Z}$ is fixed. This scheme, however, may not ensure the optimal affinity matrix since different view features characterize specific and partly independent information of the dataset. Therefore, to address this issue, different weights should be assigned on different views. Then we give the following model:

$$\begin{aligned} &\min_{\mathcal{Z},S,\omega} \|\mathcal{Z}\|_{\circledast} + \sum_{v=1}^{V} \Big(\alpha \|X^{(v)} - X^{(v)} Z^{(v)}\|_{2,1} + \\ &\qquad \lambda \omega^{(v)} \|Z^{(v)} - S\|_F^2 \Big) + \eta \|\omega\|_2^2 \\ &s.t. \quad \mathcal{Z} = \Phi(Z^{(1)}, Z^{(2)}, \cdots, Z^{(V)}), \omega \geq 0, \Sigma_v \omega^{(v)} = 1, \end{aligned} \tag{3}$$

where α, λ and η are three positive parameters to balance the contributions of all terms in the objective function; $\omega^{(v)}$ is the relative weight of the v-th view; the last term is to smoothen the weight distribution and avoid the futile solution. However, in model (3), the self-expression property is encoded on the original input data space (*i.e.*, the second term). This usually exhibits the nonlinear structure in real-world datasets. Here, we seek new feature spaces for the linear separated multi-view clustering. Borrowing the idea of the kernel methods [40, 41], for the v-th feature, let $\phi^{(v)} : \mathbb{R}^{d_v} \to \mathcal{H}^{(v)}$ be a kernel mapping from the original data space to the kernel space. As stated in the following Eq. (6), $\phi^{(v)}$ does not need to be defined explicitly. Let $\mathbf{K}^{(v)} \in \mathbb{R}^{n \times n}$ be a positive kernel Gram matrix, *i.e.*,

$$\mathbf{K}^{(v)} = \phi^{(v)}(X^{(v)})^T \phi^{(v)}(X^{(v)}). \tag{4}$$

Then, we encode the self-expression property on the new feature space. This is also the reason that the proposed JLMVC can handle the nonlinearity problem. Based on the above analysis, model (3) can be formulated as

$$\begin{aligned} &\min_{\mathcal{Z},S,\omega} \|\mathcal{Z}\|_{\circledast} + \sum_{v=1}^{V} \Big(\alpha \|\phi(X^{(v)}) - \phi(X^{(v)}) Z^{(v)}\|_{2,1} + \\ &\qquad \lambda \omega^{(v)} \|Z^{(v)} - S\|_F^2 \Big) + \eta \|\omega\|_2^2 \\ &s.t. \quad \mathcal{Z} = \Phi(Z^{(1)}, Z^{(2)}, \cdots, Z^{(V)}), \omega \geq 0, \Sigma_v \omega^{(v)} = 1. \end{aligned} \tag{5}$$

Note that the second term of Eq. (5) can be rewritten as

$$\begin{aligned} &\|\phi(X^{(v)}) - \phi(X^{(v)}) Z^{(v)}\|_{2,1} \\ &= \sum_{i=1}^{n} \left({P_i^{(v)}}^T \mathbf{K}^{(v)} P_i^{(v)} \right)^{\frac{1}{2}}, \end{aligned} \tag{6}$$

where $P^{(v)} = I - Z^{(v)}$. $P_i^{(v)}$ is the i-th column of $P^{(v)}$. From Eq. (6), it is easy to see that the kernel mapping $\phi^{(v)}$ appears only in the form of the inner product, *i.e.*, $\phi^{(v)}(X^{(v)})^T \phi^{(v)}(X^{(v)})$, leading to the kernel Gram matrix $\mathbf{K}^{(v)}$. Therefore, $\phi^{(v)}$ is implicitly defined. For simplicity, we denote $g^{(v)}(P^{(v)}) = \sum_{i=1}^{n} \left({P_i^{(v)}}^T \mathbf{K}^{(v)} P_i^{(v)} \right)^{\frac{1}{2}}$ to be the reconstruction error in the kernel space. Finally, the proposed JLMVC model can be formulated as

$$\begin{aligned} &\min_{\mathcal{Z},P^{(v)},S,\omega} \|\mathcal{Z}\|_{\circledast} + \sum_{v=1}^{V} \Big(\alpha g^{(v)}\big(P^{(v)}\big) + \\ &\qquad \lambda \omega^{(v)} \|Z^{(v)} - S\|_F^2 \Big) + \eta \|\omega\|_2^2 \\ &s.t. \quad \mathcal{Z} = \Phi(Z^{(1)}, Z^{(2)}, \cdots, Z^{(V)}), \\ &\qquad \mathcal{P} = \Phi(P^{(1)}, P^{(2)}, \cdots, P^{(V)}), \\ &\qquad \mathcal{P} = \mathcal{I} - \mathcal{Z}, \ \omega \geq 0, \ \Sigma_v \omega^{(v)} = 1, \end{aligned} \tag{7}$$

where the first term, *i.e.*, $\|\mathcal{Z}\|_{\circledast}$ defined in Eq. (1), is used to explore the low-rankness of $\mathcal{Z}$; the second term can handle the nonlinear structures; the third term with the adaptive weight strategy aims to learn a unified affinity matrix S.

B. Optimization

It is intractable to solve the proposed model in Eq. (7) since it is not jointly convex and coupled with respect to variable $\mathcal{Z}$. Therefore, we solve Eq. (7) under ADMM framework. We can reformulate Eq. (7) as:

$$\begin{aligned} &\min_{\mathcal{Z},\mathcal{Y},\mathcal{P},S,\omega} \|\mathcal{Y}\|_{\circledast} + \sum_{v=1}^{V} \Big(\alpha g^{(v)}\big(P^{(v)}\big) + \\ &\qquad \lambda \omega^{(v)} \|Z^{(v)} - S\|_F^2 \Big) + \eta \|\omega\|_2^2 \\ &s.t. \ \mathcal{Z} = \Phi(Z^{(1)}, Z^{(2)}, \cdots, Z^{(V)}), \\ &\qquad \mathcal{P} = \Phi(P^{(1)}, P^{(2)}, \cdots, P^{(V)}), \\ &\qquad \mathcal{P} = \mathcal{I} - \mathcal{Z}, \ \omega \geq 0, \ \Sigma_v \omega^{(v)} = 1, \ \mathcal{Z} = \mathcal{Y}. \end{aligned} \tag{8}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2019.2952984, IEEE Transactions on Multimedia

Following the idea of ADMM, we introduce one auxiliary variable $\mathcal{Y}$ to separate $\mathcal{Z}$ in the objective function and then iteratively update each variable by fixing other variables [45, 46]. The augmented Lagrangian function is defined as the sum of the objective function of Eq. (8) and the penalty term under the l_2-norm. The augmented Lagrangian function of model (8) is given by:

$$\mathcal{L}_\rho(\mathcal{Z},\mathcal{Y},P^{(v)},S,\omega;\Theta,\Pi)=\|\mathcal{Y}\|_\circledast+\sum_{v=1}^{V}\Big(\alpha g^{(v)}\big(P^{(v)}\big)+\lambda\omega^{(v)}\|Z^{(v)}-S\|_F^2\Big)+\eta\|\omega\|_2^2+\langle\Theta,\mathcal{I}-\mathcal{Z}-\mathcal{P}\rangle+\frac{\rho}{2}\|\mathcal{I}-\mathcal{Z}-\mathcal{P}\|_F^2+\langle\Pi,\mathcal{Z}-\mathcal{Y}\rangle+\frac{\rho}{2}\|\mathcal{Z}-\mathcal{Y}\|_F^2, \tag{9}$$

where Θ and Π are the Lagrange multipliers of size $n\times n\times V$; ρ is the non-negative penalty parameter; $\langle\cdot,\cdot\rangle$ is the inner product. Under the ADMM framework, we can solve Eq. (9) by optimizing one variable while keeping the other variables fixed as follows:

Step 1 Update $\mathcal{Z}$: Fixing other variables, we can update $\mathcal{Z}$ by the following subproblem:

$$\min_{\mathcal{Z}}\sum_{v=1}^{V}\lambda\omega_k^{(v)}\|Z^{(v)}-S_k\|_F^2+\frac{\rho_k}{2}\|\mathcal{I}-\mathcal{Z}-\mathcal{P}_k+\frac{\Theta_k}{\rho_k}\|_F^2+\frac{\rho_k}{2}\|\mathcal{Z}-\mathcal{Y}_k+\frac{\Pi_k}{\rho_k}\|_F^2. \tag{10}$$

It is easy to see that updating each frontal slice $Z^{(v)}$ of $\mathcal{Z}$ is independent. This means that $Z^{(v)}$ can be updated in parallel. The v-th subproblem is

$$\min_{Z^{(v)}}\lambda\omega_k^{(v)}\|Z^{(v)}-S_k\|_F^2+\frac{\rho_k}{2}\|Z^{(v)}-A_k^{(v)}\|_F^2+\frac{\rho_k}{2}\|Z^{(v)}-B_k^{(v)}\|_F^2, \tag{11}$$

where $A_k^{(v)}=I-P^{(v)}+\frac{\Theta_k^{(v)}}{\rho_k}$ and $B_k^{(v)}=Y_k^{(v)}-\frac{\Pi_k^{(V)}}{\rho_k}$. By setting the derivative of Eq. (11) with respect to $Z^{(v)}$ to zero, the optimal solution $Z_{k+1}^{(v)}$ is

$$Z_{k+1}^{(v)}=\big(2\lambda\omega_k^{(v)}S_k+\rho_kA_k^{(v)}+\rho_kB_k^{(v)}\big)/(2\lambda\omega_k^{(v)}+2\rho_k). \tag{12}$$

Step 2 Update $\mathcal{Y}$: When other variables are fixed, $\mathcal{Y}$ can be updated by

$$\min_{\mathcal{Y}}\|\mathcal{Y}\|_\circledast+\frac{\rho_k}{2}\|\mathcal{Y}-\mathcal{F}_k\|_F^2, \tag{13}$$

where $\mathcal{F}_k=\mathcal{Z}_{k+1}+\frac{\Pi_k}{\rho_k}$. Following [7], we rotate $\mathcal{Y}$ from size $n\times n\times V$ to $n\times V\times n$ as shown in Fig. 3. The first reason is that, as in Eq. (1), t-SVD-TNN performs SVD on each frontal slice of $\hat{\mathcal{Y}}$ to capture the "spatial-shifting" correlation [44, 47]. This means that t-SVD-TNN preserves only the low-rank property of intra-view. However, we hope to capture the low-rank property of inter-views. The second reason is that the rotation operation can significantly reduce the computation cost [7]. After the rotation operation, each frontal slice of $\hat{\mathcal{Y}}$ represents the view-specific self-representation matrix.

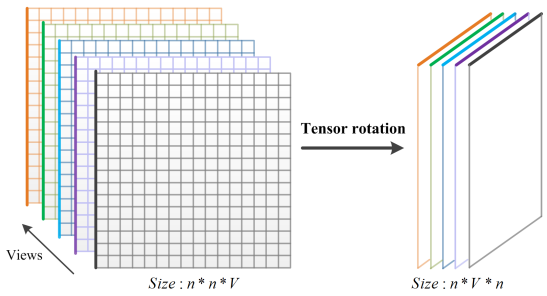

Fig. 3. Explanation of rotation.

The closed-form solution of Eq. (13) can be obtained by the tensor tubal-shrinkage operator [7, 48]:

$$\mathcal{Y}_{k+1}=\mathcal{C}_{\frac{V}{\rho_k}}(\mathcal{F}_k)=\mathcal{U}*\mathcal{C}_{\frac{V}{\rho_k}}(\mathcal{G})*\mathcal{V}^T, \tag{14}$$

where $\mathcal{F}_k=\mathcal{U}*\mathcal{G}*\mathcal{V}^T$, and $\mathcal{C}_{\frac{V}{\rho_k}}(\mathcal{G})=\mathcal{G}*\mathcal{J}$, in which $\mathcal{J}$ is an f-diagonal tensor whose diagonal element in the Fourier domain is $\mathcal{J}(i,i,k)=\max\{1-\frac{V/\rho_k}{\mathcal{G}(i,i,k)},0\}$.

Step 3 Update $\mathcal{P}$: With other variables fixed, we minimize the augmented Lagrangian function in Eq. (9) with respect to $\mathcal{P}$:

$$\min_{\mathcal{P}}\sum_{v=1}^{V}\alpha g^{(v)}\big(P^{(v)}\big)+\frac{\rho_k}{2}\|\mathcal{I}-\mathcal{Z}_{k+1}-\mathcal{P}+\frac{\Theta_k}{\rho_k}\|_F^2. \tag{15}$$

Similar to Eq. (10), updating $P^{(v)}$ is also independent:

$$\min_{P^{(v)}}\alpha g^{(v)}\big(P^{(v)}\big)+\frac{\rho_k}{2}\|P^{(v)}-D_k^{(v)}\|_F^2, \tag{16}$$

where $D_k^{(v)}=I-Z_{k+1}^{(v)}+\frac{\Theta_k^{(v)}}{\rho_k}$. Compared with the method in [40] which uses l_2-norm to measure the reconstruction error, it is more difficult to solve Eq. (16) since $g^{(v)}$ is convex but non-smooth. According to [41], the i-th column of the optimal solution of Eq. (16) $p_i^{(v)}$ is

$$p_i^{(v)}=\begin{cases}\hat{p}^{(v)}, & \text{if } \|[1/\sigma_1^{(v)},\cdots,1/\sigma_r^{(v)}]\circ t_u^{(v)}\|>1/\tau;\\ c_i^{(v)}-V_{\mathbf{K}}^{(v)}t_u^{(v)}, & \text{otherwise.}\end{cases} \tag{17}$$

where $\tau=\frac{\rho_k}{\alpha}$; $\circ$ is the element multiplication operator; $\mathbf{K}^{(v)}=V^{(v)}\Sigma^{(v)^2}V^{(v)^T}$ is the singular value decomposition of $\mathbf{K}^{(v)}$; $\Sigma^{(v)}=\text{diag}(\sigma_1^{(v)},\cdots,\sigma_r^{(v)},0,\cdots,0)$ and r is the rank of $\mathbf{K}^{(v)}$; $V_{\mathbf{K}}^{(v)}$ is constructed by the first r columns of $V^{(v)}$; $t_u^{(v)}=V_{\mathbf{K}}^{(v)}c_i^{(v)}$; $\hat{p}^{(v)}$ is defined as

$$\hat{p}^{(v)}=c_i^{(v)}-V_{\mathbf{K}}^{(v)}\big([\frac{\sigma_1^{(v)2}}{\gamma^{(v)}+\sigma_1^{(v)2}},\cdots,\frac{\sigma_r^{(v)2}}{\gamma^{(v)}+\sigma_r^{(v)2}}]^T\circ t_u^{(v)}\big), \tag{18}$$

where $\gamma^{(v)}>0$ is a scalar, and it satisfies

$$t_u^{(v)^T}\text{diag}(\{\frac{\sigma_i^{(v)2}}{(\gamma^{(v)}+\sigma_i^{(v)2})^2}\}_{1\le i\le r})t_u^{(v)}=1/\tau^2. \tag{19}$$

We can obtain a unique root $\gamma^{(v)}$ when $\|[1/\sigma_1^{(v)},\cdots,1/\sigma_r^{(v)}]\circ t_u^{(v)}\|>1/\tau$.

Step 4 Update S: When keeping other variables fixed, we obtain the following optimization problem:

$$\begin{aligned} S_{k+1} &= \arg\min_{S} \sum_{v=1}^{V} \omega_k^{(v)} \|Z_{k+1}^{(v)} - S\|_F^2, \\ &= \sum_{v=1}^{V} \omega_k^{(v)} Z_{k+1}^{(v)}. \end{aligned} \tag{20}$$

The last equation is based on the fact that $\sum_v \omega_k^{(v)} = 1$.

Step 5 Update ω: To obtain the adaptive weights ω_{k+1}, we minimize the augmented Lagrangian function in Eq. (9) with respect to ω:

$$\begin{aligned} \omega_{k+1} = \arg\min_{\omega} \sum_{v=1}^{V} \omega^{(v)} \|Z_{k+1}^{(v)} - S_{k+1}\|_F^2 + \eta\|\omega\|_2^2, \\ s.t.\ \omega \geq 0,\ \sum_v \omega^{(v)} = 1. \end{aligned} \tag{21}$$

Actually, Eq. (21) is a quadratic programming problem

$$\begin{aligned} \omega_{k+1} = \arg\min_{\omega} \|\omega + \frac{g_k}{2\eta}\|_2^2, \\ s.t.\ \omega \geq 0,\ \sum_v \omega_v = 1. \end{aligned} \tag{22}$$

where $g_k^v = \|Z_{k+1}^{(v)} - S_{k+1}\|_F^2$ forms the vector g_k. We adopt the off-the-shelf quadratic programming solver to solve the above problem.

Step 6 Update Θ, Π, and ρ: The Lagrangian multipliers Θ, Π and the penalty parameter ρ are updated by

$$\begin{aligned} \Theta_{k+1} &= \Theta_k + \rho_k(\mathcal{I} - \mathcal{Z}_{k+1} - \mathcal{P}_{k+1}); \\ \Pi_{k+1} &= \Pi_k + \rho_k(\mathcal{Z}_{k+1} - \mathcal{Y}_{k+1}); \\ \rho_{k+1} &= \min\{\beta * \rho_k, \rho_{max}\}, \end{aligned} \tag{23}$$

where $\beta \in [0, \frac{\sqrt{5}+1}{2}]$ is a step length to update the penalty parameter ρ in each iteration [49]. ρ_{max} is the maximum value of the penalty parameter ρ.

The details of the proposed algorithm for solving the JLMVC model are summarized in Algorithm 1. Algorithm 1 can be terminated when the following convergence condition is satisfied

$$\max \left\{ \begin{array}{l} \|I - Z_{k+1}^{(v)} - P_{k+1}^{(v)}\|_\infty, v = 1, \cdots, V \\ \|\mathcal{Z}_{k+1} - \mathcal{Y}_{k+1}\|_\infty \end{array} \right\} \leq tol, \tag{24}$$

where $tol > 0$ is a pre-defined tolerance.

Several notes regarding Algorithm 1 are given below to further understand the proposed JLMVC.

- The weights of different views are of importance to the construction of the affinity matrix. An intuitive way to initialize weights of different views is set each weight to be $\omega_1^{(v)} = \frac{1}{V}$. Then, weights are updated in an adaptive manner by Eq. (22). Other variables $\mathcal{Y}_1$, $\mathcal{Z}_1$, S_1, Θ_1, Π_1 are initialized to $\mathbf{0}$.
- Lines 3-6 of Algorithm 1 can be performed in parallel as subproblems (11) and (16) are independent with respect to $Z^{(v)}$ and $P^{(v)}$, respectively.
- After performing Algorithm 1, we can obtain the unified affinity matrix S which well inherits the advantage of the representation tensor $\mathcal{Z}$. Finally, the learned affinity matrix S serves as the input of spectral clustering algorithm [9] to yield the clustering results.

Algorithm 1 JLMVC for multi-view clustering

Input: multi-view features: $\{X^{(v)}\}$; parameters: α, λ;
Initialize: $\mathcal{Y}_1$, $\mathcal{Z}_1$, S_1, Θ_1, Π_1 initialized to $\mathbf{0}$; weight $\omega_1^{(v)} = \frac{1}{V}$; $\eta = 500$, $\rho_1 = 10^{-3}$, $\beta = 1.5$, $\epsilon = 10^{-7}$, $k = 1$;
1: Calculate the v-th kernel matrix $\mathbf{K}^{(v)}$ by Eq. (4) ($v = 1, \cdots, V$);
2: **while** not converged **do**
3: **for** $v = 1$ to V **do**
4: Update $Z_{k+1}^{(v)}$ by Eq. (12);
5: Update $P_{k+1}^{(v)}$ by Eq. (17);
6: **end for**
7: Update $\mathcal{Y}_{k+1}$ by Eq. (14);
8: Update S_{k+1} by Eq. (20);
9: Update ω_{k+1} by Eq. (22);
10: Update Θ_{k+1}, Π_{k+1}, and ρ_{k+1} by Eq. (23);
11: Check the convergence condition in Eq. (24);
12: **end while**
Output: Affinity matrix S_{k+1}.

IV. Experimental results

In this section, we aim to evaluate the performance of JLMVC on eight multimedia datasets. The model analysis is also reported.

A. Experimental settings

Our experiments select eight multimedia datasets for multi-view clustering, including four face image datasets, two scene datasets, one prokaryotic dataset, and one article data. The details of each dataset are listed as follows:

Dataset descriptions: Yale (http://cvc.yale.edu/projects/yalefaces/yalefaces.html): it consists of 165 gray-scale images of 15 individuals with different facial expressions and configurations. Following [6, 7], $4096d$ (dimension, d) Intensity, $3304d$ LBP, and $6750d$ Gabor are extracted as three multi-view features; **Extended YaleB** (http://vision.ucsd.edu/~leekc/ExtYaleDatabase/ExtYaleB.html): it contains 2414 face images of 38 individuals, each of which has 64 near frontal images under different lighting conditions. Similar to [6, 7], the first 10 classes are selected and three types of features, including $2500d$ Intensity, $3304d$ LBP, and $6750d$ Gabor, are extracted; **ORL** (http://www.uk.research.att.com/facedatabase.html): it includes 400 face images with 40 clusters under different times, lighting, facial expressions, and facial details; **Prokaryotic phyla:** it contains 551 prokaryotic species described by textual data and different genomic representations. **Wikipedia** (http://lig-membres.imag.fr/grimal/data.html): it is an article dataset selected by Wikipedia editors since 2009. 693 documents with 2 views are selected; **COIL-20** (http://www.cs.columbia.edu/CAVE/software/softlib/): COIL_20 contains 1440 images of 20 object categories. Three view features

TABLE II
CLUSTERING RESULTS (MEAN±STANDARD DEVIATION) ON THREE FACE IMAGE DATASETS.

Data	Method	ACC	NMI	AR	F-score	Precision	Recall
Extended YaleB ($\alpha = 0.3,\ \lambda = 0.1$)	SSC_{best}	0.587±0.003	0.534±0.003	0.430±0.005	0.487±0.004	0.451±0.002	0.509±0.007
	LRR_{best}	0.615±0.013	0.627±0.040	0.451±0.002	0.508±0.004	0.481±0.002	0.539±0.001
	MLAP	0.278±0.002	0.231±0.002	0.119±0.002	0.207±0.001	0.204±0.001	0.211±0.001
	DiMSC	0.615±0.003	0.636±0.002	0.453±0.005	0.504±0.006	0.481±0.004	0.534±0.004
	LT-MSC	0.626±0.010	0.637±0.003	0.459±0.030	0.521±0.006	0.485±0.001	0.539±0.002
	MLAN	0.346±0.011	0.352±0.015	0.093±0.009	0.213±0.023	0.159±0.018	0.321±0.013
	ECMSC	0.783±0.011	0.759±0.012	0.544±0.008	0.597±0.010	0.513±0.009	0.718±0.006
	t-SVD-MSC	0.652±0.000	0.667±0.004	0.500±0.003	0.550±0.002	0.514±0.004	0.590±0.004
	HLR-M^2VS	0.670±0.002	0.703±0.006	0.529±0.006	0.577±0.003	0.560±0.001	0.595±0.001
	Kt-SVD-MSC	0.896±0.016	0.893±0.015	0.813±0.027	0.832±0.024	0.821±0.024	0.842±0.024
	DMF-MVC	0.763±0.001	0.649±0.002	0.512±0.002	0.564±0.001	0.525±0.001	0.610±0.001
	AWP	0.697±0.000	0.715±0.000	0.517±0.000	0.548±0.000	0.520±0.000	0.579±0.000
	JLMVC	**0.910**±0.022	**0.897**±0.010	**0.832**±0.019	**0.849**±0.017	**0.837**±0.019	**0.860**±0.015
Yale ($\alpha = 0.7,\ \lambda = 0.05$)	SSC_{best}	0.627±0.000	0.671±0.011	0.475±0.004	0.517±0.004	0.509±0.003	0.547±0.004
	LRR_{best}	0.697±0.001	0.709±0.011	0.512±0.005	0.547±0.007	0.529±0.005	0.567±0.004
	MLAP	0.727±0.010	0.751±0.014	0.580±0.021	0.606±0.020	0.589±0.020	0.624±0.036
	DiMSC	0.709±0.003	0.727±0.010	0.535±0.003	0.564±0.010	0.543±0.012	0.586±0.009
	LT-MSC	0.741±0.002	0.765±0.008	0.570±0.004	0.598±0.006	0.569±0.004	0.629±0.005
	MLAN	0.558±0.008	0.590±0.004	0.273±0.008	0.330±0.002	0.257±0.004	0.463±0.006
	ECMSC	0.771±0.014	0.773±0.010	0.590±0.014	0.617±0.012	0.584±0.013	0.653±0.013
	t-SVD-MSC	0.963±0.006	0.953±0.008	0.910±0.010	0.915±0.007	0.904±0.005	0.927±0.007
	HLR-M^2VS	0.756±0.014	0.790±0.008	0.631±0.016	0.654±0.014	0.627±0.022	0.684±0.010
	Kt-SVD-MSC	0.982±0.000	0.987±0.000	0.973±0.000	0.975±0.000	0.971±0.000	0.979±0.000
	DMF-MVC	0.745±0.011	0.782±0.010	0.579±0.002	0.601±0.002	0.598±0.001	0.613±0.002
	AWP	0.715±0.000	0.738±0.000	0.556±0.000	0.584±0.000	0.562±0.000	0.609±0.000
	JLMVC	**1.000**±0.000	**1.000**±0.000	**1.000**±0.000	**1.000**±0.000	**1.000**±0.000	**1.000**±0.000
ORL ($\alpha = 0.05,\ \lambda = 0.005$)	SSC_{best}	0.765±0.008	0.893±0.007	0.694±0.013	0.682±0.012	0.673±0.007	0.764±0.005
	LRR_{best}	0.773±0.003	0.895±0.006	0.724±0.020	0.731±0.004	0.701±0.001	0.754±0.002
	MLAP	0.789±0.021	0.895±0.010	0.714±0.025	0.720±0.024	0.686±0.027	0.759±0.024
	DiMSC	0.838±0.001	0.940±0.003	0.802±0.000	0.807±0.003	0.764±0.012	0.856±0.004
	LT-MSC	0.795±0.007	0.930±0.003	0.750±0.003	0.768±0.004	0.766±0.009	0.837±0.005
	MLAN	0.705±0.022	0.854±0.018	0.384±0.010	0.376±0.015	0.254±0.021	0.721±0.020
	ECMSC	0.854±0.011	0.947±0.009	0.810±0.012	0.821±0.015	0.783±0.008	0.859±0.012
	t-SVD-MSC	0.970±0.003	0.993±0.002	0.967±0.002	0.968±0.003	0.946±0.004	0.991±0.003
	HLR-M^2VS	**1.000**±0.000	**1.000**±0.000	**1.000**±0.000	**1.000**±0.000	**1.000**±0.000	**1.000**±0.000
	Kt-SVD-MSC	0.971±0.021	0.994±0.007	0.972±0.022	0.972±0.022	0.956±0.027	0.991±0.017
	DMF-MVC	0.805±0.006	0.891±0.004	0.722±0.010	0.728±0.010	0.704±0.011	0.755±0.009
	AWP	0.753±0.000	0.908±0.000	0.697±0.000	0.705±0.000	0.615±0.000	0.824±0.000
	JLMVC	0.983±0.018	0.996±0.004	0.983±0.018	0.984±0.017	0.973±0.028	0.994±0.006

including $1024d$ intensity, $3304d$ LBP, and $6750d$ Gabor are employed; **CMU-PIE** (http://vasc.ri.cmu.edu/idb/html/face/): it consists of 5440 facial images of 68 subjects. Each image is of size 64×64 with a large variance. Following [50], three types of features including $1024d$ Intensity, $256d$ LBP, and $496d$ HOG are used; **Scene-15** [51]: it contains 4485 outdoor and indoor scene images from 15 categories. Following [7], three kinds of image features, including $1800d$ PHOW, $1180d$ PRI-CoLBP, and $1240d$ CENTRIST are extracted to represent Scene-15.

Baselines: Our proposed JLMVC is compared with twelve state-of-the-art single-view and multi-view clustering methods. The competing methods are listed as follows: $\textbf{SSC}_{\textbf{best}}$ [20]: single-view clustering using the sparse regularizer (l_1-norm) to construct the representation matrix; $\textbf{LRR}_{\textbf{best}}$ [19]: single-view clustering using the nuclear norm to construct the representation matrix; **MLAP** [33]: multi-view clustering by concatenating representation matrices of different views and imposing low-rank constraint to explore the complementarity; **DiMSC** [52]: multi-view clustering with the Hilbert-Schmidt Independence criterion; **LT-MSC** [6]: multi-view clustering with the low-rank tensor constraint; **MLAN** [16]: multi-view clustering with adaptive neighbors; **ECMSC** [23]: multi-view clustering by simultaneously exploiting the representation exclusivity and indicator consistency; **t-SVD-MSC** [7]: multi-view clustering via tensor multi-rank minimization; **HLR-M^2VS** [8]: multi-view clustering via hyper-Laplacian regularized multilinear multiview self-representations; **Kt-SVD-MSC** [43]: multi-view clustering via robust kernelized multi-view self-representations; **DMF-MVC** [29]: multi-view clustering via deep matrix factorization; **AWP** [53]: multi-view clustering via adaptively weighted procrustes.

Specifically, $\text{SSC}_{\textbf{best}}$ and $\text{LRR}_{\textbf{best}}$ are two representative baselines for single-view clustering. Others are the multi-view clustering baselines. LT-MSC, t-SVD-MSC, HLR-M^2VS, and Kt-SVD-MSC are low-rank tensor representation-based multi-view clustering approaches. Kt-SVD-MSC is the kernelized version of t-SVD-MSC. MLAN is graph-based multi-view

TABLE III
CLUSTERING RESULTS (MEAN±STANDARD DEVIATION) ON WIKIPEDIA AND PROKARYOTIC.

Data	Method	ACC	NMI	AR	F-score	Precision	Recall
Wikipedia ($\alpha = 0.001,\ \lambda = 0.55$)	SSC_{best}	0.561±0.001	0.527±0.002	0.418±0.001	0.481±0.001	0.491±0.001	0.471±0.001
	LRR_{best}	0.554±0.001	0.523±0.001	0.417±0.000	0.479±0.000	0.490±0.000	0.468±0.001
	MLAP	0.574±0.000	0.510±0.000	0.414±0.000	0.477±0.000	0.482±0.000	0.472±0.000
	DiMSC	0.547±0.007	0.500±0.003	0.397±0.002	0.461±0.002	0.478±0.002	0.445±0.002
	LT-MSC	0.532±0.003	0.496±0.005	0.407±0.005	0.471±0.005	0.480±0.004	0.461±0.006
	MLAN	0.203±0.001	0.066±0.000	0.020±0.000	0.127±0.000	0.127±0.000	0.127±0.000
	ECMSC	0.563±0.000	0.522±0.000	0.413±0.000	0.475±0.000	0.494±0.000	0.457±0.000
	t-SVD-MSC	0.527±0.011	0.480±0.001	0.393±0.002	0.458±0.002	0.470±0.002	0.447±0.002
	HLR-M^{2}VS	0.577±0.000	0.513±0.000	0.417±0.000	0.480±0.000	0.485±0.000	0.475±0.000
	Kt-SVD-MSC	0.573±0.002	0.538±0.002	0.443±0.002	0.502±0.002	0.513±0.002	0.492±0.002
	AWP	0.573±0.000	0.543±0.000	0.434±0.000	0.497±0.000	0.493±0.000	0.501±0.000
	JLMVC	**0.587**±0.000	**0.552**±0.000	**0.462**±0.000	**0.520**±0.000	**0.527**±0.000	**0.513**±0.000
Prokaryotic ($\alpha = 0.001,\ \lambda = 0.1$)	SSC_{best}	0.466±0.000	0.242±0.000	0.083±0.000	0.439±0.000	0.446±0.000	0.432±0.000
	LRR_{best}	0.499±0.000	0.245±0.000	0.115±0.000	0.410±0.000	0.485±0.000	0.355±0.000
	MLAP	0.583±0.000	0.243±0.000	0.203±0.000	0.479±0.000	0.546±0.000	0.436±0.000
	DiMSC	0.395±0.001	0.070±0.000	0.053±0.000	0.346±0.000	0.441±0.000	0.284±0.000
	LT-MSC	0.431±0.007	0.156±0.020	0.051±0.016	0.401±0.006	0.429±0.011	0.376±0.003
	MLAN	0.712±0.002	0.387±0.003	0.425±0.003	0.618±0.002	0.728±0.002	0.537±0.002
	ECMSC	0.432±0.001	0.193±0.001	0.078±0.001	0.383±0.002	0.457±0.002	0.329±0.001
	t-SVD-MSC	0.523±0.000	0.197±0.000	0.137±0.000	0.486±0.000	0.474±0.000	0.500±0.000
	HLR-M^{2}VS	0.646±0.002	0.332±0.001	0.288±0.001	0.533±0.001	0.611±0.001	0.473±0.001
	Kt-SVD-MSC	0.744±0.002	0.475±0.002	0.471±0.002	0.646±0.001	0.769±0.001	0.557±0.001
	AWP	0.603±0.000	0.342±0.000	0.301±0.000	0.518±0.000	0.657±0.000	0.428±0.000
	JLMVC	**0.766**±0.001	**0.487**±0.002	**0.500**±0.001	**0.670**±0.002	**0.775**±0.001	**0.591**±0.001

DMF-MVCC was crashed on these two databases.

clustering one. The source codes of all competing methods are downloaded from the authors' homepages. For single-view clustering methods, we perform SSC and LRR on each feature matrix independently and report the best clustering results. For multi-view clustering ones, LT-MSC, t-SVD-MSC, HLR-M^{2}VS, and Kt-SVD-MSC are first performed to learn the representation tensor $\mathcal{Z}$, and then conduct the affinity matrix S by averaging each frontal slice of $\mathcal{Z}$, that is, $S = \frac{1}{V}\sum_v \left(|Z^{(v)}| + |Z^{(v)^T}|\right)$. This means that they are performed in two separate steps to obtain the affinity matrix. After that, the spectral clustering algorithm [9] is carried out to obtain the final clustering results. For fair comparison, our experiments follow the same parameter settings of the original papers. For SSC and LRR, we select the regularization parameter from the interval $[0.01,\ 10]$; for MLAP, two free parameters are searched from 0.001 to 1; for DiMSC, two free parameters are chosen from $[0.01, 0.03]$ and $[20 : 20 : 180]$, respectively; the trade-off parameter of LT-MSC is selected from 0.01 to 100; for MLAN, one parameter is set to a random number between 1 and 30; three free parameters of ECMSC are set in $[0.1, 1]$, $[0.1, 1]$, and 1.2, respectively; the trade-off parameters of t-SVD-MSC and Kt-SVD-MSC are set within the range $[0.1, 2]$ and $[0.001, 0.6]$, respectively; for HLR-M^{2}VS, two parameters are located within the ranges $[0.01, 0.2]$ and $[0.1, 0.9]$, respectively; DMF-MVC adopts $\{[100, 50],\ [500, 50],\ [500, 200]\}$ as the sizes of the last layer and other parameters use the default settings as recommended in [29]; AWP is parameter-free.

Evaluation metrics: Six widely used metrics are selected to evaluate the clustering quality including accuracy (ACC), normalized mutual information (NMI), adjusted rank index (AR), F-score, Precision, and Recall. For each evaluation metric, the higher value indicates the better clustering performance. As we know, the spectral clustering algorithm uses the K-means algorithm to obtain the indicator matrix for all methods except MLAN, and different initializations may yield different clustering results. Thus, we run 10 trials for each experiment on all datasets and report their average performance with standard deviations. Although MLAN does not use the K-means algorithm, there exists one random parameter. Thus, we repeat MLAN algorithm 10 trials.

B. Clustering performance comparison

The clustering performance comparison on all multimedia datasets are reported in Tables II, III, and IV. The best results are highlighted in bold and the second-best ones are underlined in each table. From the results in these tables, we reach the following conclusions:

- Generally speaking, the proposed JLMVC achieves the best results on all datasets, except the ORL data where JLMVC is the second best. They have verified the validity of the proposed JLMVC. This is mainly because the proposed JLMVC takes three aspects into one unified model: 1) high correlation between the representation tensor and affinity matrix; (2) the nonlinear structures in real applications; (3) different contributions of each view for the construction of the unified affinity matrix. (More details can be found in Section IV-C-(3).) Take the Extended YaleB data as an example, the proposed JLMVC improves around 1.4%, 0.4%, 2.1%, 1.7%, 1.6%, and 1.8% with respect to six measures over the second-best method Kt-SVD-MSC which also exploits the kernel trick to solve the nonlinear subspaces problem but learns the

TABLE IV
CLUSTERING RESULTS (MEAN±STANDARD DEVIATION) ON *COIL-20*, *CMU-PIE* AND *Scene-15*.

Data	Method	ACC	NMI	AR	F-score	Precision	Recall
COIL-20 ($\alpha = 0.001,\ \lambda = 0.001$)	$\text{SSC}_{\mathbf{best}}$	0.803±0.022	0.935±0.009	0.798±0.022	0.809±0.013	0.734±0.027	0.804±0.028
	$\text{LRR}_{\mathbf{best}}$	0.761±0.003	0.829±0.006	0.720±0.020	0.734±0.006	0.717±0.003	0.751±0.002
	MLAP	0.738±0.020	0.825±0.009	0.685±0.023	0.701±0.021	0.688±0.027	0.715±0.016
	DiMSC	0.778±0.022	0.846±0.002	0.732±0.005	0.745±0.005	0.739±0.007	0.751±0.003
	LT-MSC	0.804±0.011	0.860±0.002	0.748±0.004	0.760±0.007	0.741±0.009	0.776±0.006
	MLAN	0.862±0.011	0.961±0.004	0.835±0.006	0.844±0.013	0.758±0.008	**0.953**±0.007
	ECMSC	0.782±0.001	0.942±0.001	0.781±0.001	0.794±0.001	0.695±0.002	0.925±0.001
	t-SVD-MSC	0.830±0.000	0.884±0.005	0.786±0.003	0.800±0.004	0.785±0.007	0.808±0.001
	HLR-M^{2}VS	0.852±0.009	0.960±0.006	0.833±0.005	0.842±0.003	0.757±0.010	0.949±0.011
	Kt-SVD-MSC	0.940±0.008	0.967±0.005	0.928±0.012	0.932±0.011	0.930±0.013	0.934±0.010
	DMF-MVC	0.839±0.009	0.843±0.009	**0.951**±0.001	0.852±0.008	0.786±0.015	0.843±0.001
	AWP	0.650±0.000	0.909±0.000	0.695±0.000	0.714±0.000	0.573±0.000	0.949±0.000
	JLMVC	**0.945**±0.037	**0.970**±0.024	0.937±0.033	**0.940**±0.042	**0.940**±0.043	0.941±0.042
CMU-PIE ($\alpha = 0.1,\ \lambda = 0.005$)	$\text{SSC}_{\mathbf{best}}$	0.495±0.011	0.694±0.006	0.259±0.013	0.273±0.013	0.207±0.015	0.403±0.007
	$\text{LRR}_{\mathbf{best}}$	0.527±0.009	0.673±0.011	0.330±0.014	0.341±0.017	0.290±0.016	0.415±0.011
	MLAP	0.404±0.015	0.583±0.011	0.280±0.011	0.291±0.011	0.276±0.010	0.309±0.013
	DiMSC	0.521±0.018	0.652±0.014	0.357±0.012	0.401±0.019	0.311±0.013	0.384±0.011
	LT-MSC	0.602±0.010	0.725±0.007	0.455±0.013	0.464±0.013	0.425±0.017	0.510±0.010
	MLAN	0.324±0.000	0.492±0.000	0.019±0.000	0.047±0.000	0.024±0.000	0.526±0.000
	ECMSC	0.387±0.001	0.577±0.001	0.224±0.003	0.237±0.002	0.207±0.002	0.275±0.001
	t-SVD-MSC	0.857±0.012	0.919±0.006	0.771±0.012	0.775±0.011	0.719±0.021	0.840±0.009
	HLR-M^{2}VS	0.770±0.011	0.852±0.010	0.670±0.018	0.675±0.017	0.627±0.020	0.732±0.019
	Kt-SVD-MSC	0.901±0.010	0.964±0.005	0.888±0.011	0.889±0.011	0.865±0.012	0.915±0.011
	DMF-MVC	0.534±0.015	0.707±0.006	0.360±0.018	0.371±0.017	0.309±0.021	0.465±0.009
	AWP	0.408±0.000	0.622±0.000	0.225±0.000	0.240±0.000	0.183±0.000	0.351±0.000
	JLMVC	**0.918**±0.014	**0.970**±0.003	**0.904**±0.011	**0.905**±0.011	**0.881**±0.016	**0.932**±0.007
Scene-15 ($\alpha = 0.001,\ \lambda = 1$)	$\text{SSC}_{\mathbf{best}}$	0.444±0.003	0.470±0.002	0.279±0.001	0.337±0.002	0.292±0.001	0.397±0.001
	$\text{LRR}_{\mathbf{best}}$	0.445±0.013	0.426±0.018	0.272±0.015	0.324±0.010	0.316±0.0105	0.333±0.015
	MLAP	0.568±0.005	0.563±0.002	0.405±0.002	0.447±0.002	0.439±0.001	0.455±0.003
	DiMSC	0.300±0.010	0.269±0.009	0.117±0.012	0.181±0.010	0.173±0.016	0.190±0.010
	LT-MSC	0.574±0.009	0.571±0.011	0.424±0.010	0.465±0.007	0.452±0.003	0.479±0.008
	MLAN	0.331±0.000	0.475±0.000	0.151±0.000	0.248±0.000	0.150±0.000	0.731±0.000
	ECMSC	0.457±0.001	0.463±0.002	0.303±0.001	0.357±0.001	0.318±0.001	0.408±0.001
	t-SVD-MSC	0.812±0.007	0.858±0.007	0.771±0.003	0.788±0.001	0.743±0.006	0.839±0.003
	HLR-M^{2}VS	0.878±0.003	0.895±0.005	0.850±0.003	0.861±0.005	0.850±0.008	0.871±0.010
	Kt-SVD-MSC	0.984±0.000	0.966±0.000	0.967±0.000	0.969±0.000	0.971±0.000	0.968±0.000
	DMF-MVC	0.526±0.004	0.525±0.002	0.369±0.001	0.414±0.004	0.399±0.005	0.430±0.004
	AWP	0.574±0.000	0.577±0.000	0.412±0.000	0.460±0.000	0.394±0.000	0.551±0.000
	JLMVC	**0.988**±0.000	**0.975**±0.000	**0.975**±0.000	**0.977**±0.000	**0.979**±0.000	**0.975**±0.000

representation tensor and affinity matrix in two separate manners;

- The low-rank tensor representation-based MVC methods (LT-MSC, t-SVD-MSC, HLR-M^{2}VS, Kt-SVD-MSC, and the proposed JLMVC) show better results than all single-view clustering methods (SSC and LRR) in most cases. This is mostly due to the fact that different features characterize different and partly independent information of the datasets. LRR and SSC exploit only partial information, leading to unsatisfactory results especially when multi-view features are heterogeneous. Whereas, the low-rank tensor representation-based MVC can well explore the high order correlations underlying multi-view features;
- The graph-based multi-view clustering method, MLAN, obtains unstable results. On Prokaryotic data, MLAN achieves the similar performance with our JLMVC. However, it performs worse than those single-view clustering methods on other datasets. The reason may be that the graph-based clustering approaches usually construct the affinity matrix on the raw multimedia features which may be corrupted by noise and outliers;
- On ORL data, HLR-M^{2}VS achieves better results than the proposed JLMVC. The reason is that the manifold regularization may be better to preserve the local geometrical structure of ORL data than the kernel trick when handling nonlinearity. However, HLR-M^{2}VS is less robust on Yale and Extended YaleB datasets. Specifically, in terms of ACC and NMI, the leading margins of our JLMVC are 24.0% and 19.4% over HLR-M^{2}VS on Extended YaleB, respectively. On Yale, the improvement of JLMVC is 24.4% and 21.0%, respectively. Similar observations can be obtained on Scene-15 and Prokaryotic datasets. This indicates that, compared to the manifold-based methods, the kernel-based methods may be a better way to handle the nonlinear subspaces;
- The performance of MLAP degrades sharply on the Extended YaleB data. Its performance is even worse

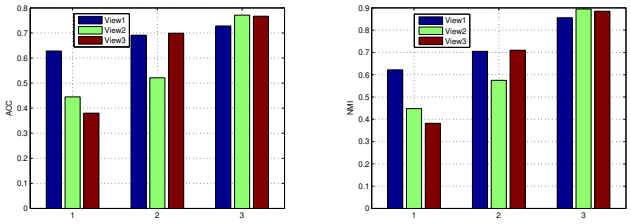

Fig. 4. ACC and NMI values of LRR with all features on (1) Extended YaleB, (2) Yale and (3) ORL datasets.

than those of the single-view clustering methods, *i.e.*, LRR and SSC. However, it performs better than them on other datasets. As stated in [7], the LBP and Gabor features cause less discriminative representation than the intensity feature due to large variations of illumination as shown in the first group of Fig. 4. This indicates that simply concatenating all features may fail to obtain a good affinity matrix to describe the relationship among all samples, especially when all features are heterogeneous. This is the direct motivation why our model considers different contributions of different features to construct the affinity matrix.

C. Model analysis

In this section, we aim to give a comprehensive analysis of the proposed JLMVC in Eq. (7), including the parameter analysis, convergence analysis, and runtime.

(1) Parameter analysis: There are three parameters, *i.e.*, α, λ, η in the proposed JLMVC. In all experiments, we set $\eta = 500$. Thus, there are two free parameters which need to be tuned. Actually, α and λ are used to balance the contributions of the low-rank tensor term, noise term and consensus term. For example, when the noise level of features is high, α may be selected a large value. α and λ are selected from the ranges $[0.001, 0.005, 0.01, 0.05, 0.1, 0.3, 0.5, 0.7]$ and $[0.001, 0.005, 0.01, 0.05, 0.1, 0.3, 0.5, 0.7, 0.9, 1]$, respectively. Here, the Yale and Extended YaleB datasets are selected as two examples. Fig. 5 shows the ACC and NMI values with respect to different combinations of α and λ. From this figure, we can observe that when α is set to a relatively large value, JLMVC can achieve the best results. An intuitive interpretation is that there are large variations of illumination on the Extended YaleB data.

(2) Computation complexity and empirical convergence analysis: The proposed JLMVC consists of six subproblems. The main computation complexity of JLMVC is to update $\mathcal{Y}$ and $\mathcal{P}$ since updating other variables contains only the matrix addition and scalar-matrix multiplication. The total computation complexity of $\mathcal{Y}$ subproblem is $\mathcal{O}(2Vn^2log(n) + V^2n^2)$ since it needs to compute the FFT, inverse FFT and singular value decomposition. For updating $\mathcal{P}$, it includes V independent subproblems as shown in Eq. (16). Each subproblem takes $\mathcal{O}(rn^2)$ for the vector-matrix multiplication, where r is the rank of $\mathbf{K}^{(v)}$. Thus, the computation complexity of JLMVC is $\mathcal{O}(2Vn^2log(n) + V^2n^2 + Vrn^2)$.

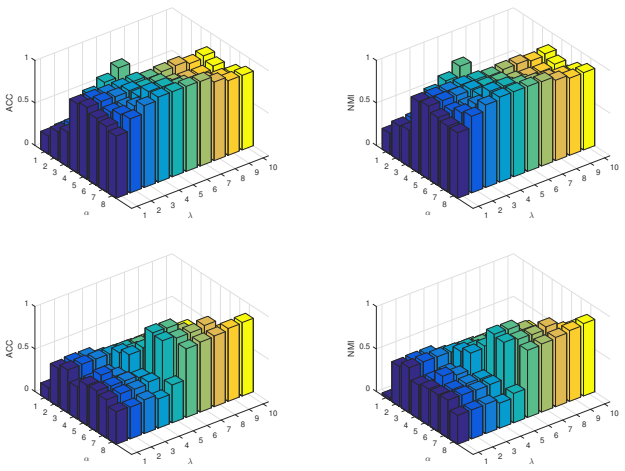

Fig. 5. ACC and NMI values of JLMVC with different combinations of α and λ on Yale (Two top figures) and Extended YaleB (Two bottom figures) datasets.

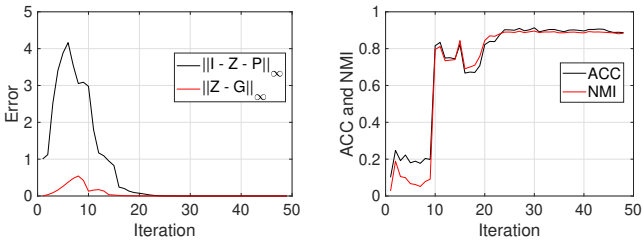

Fig. 6. Empirical convergence (Left), ACC and NMI values (Right) versus iterations on Extended YaleB data.

The empirical convergence of JLMVC on Extended YaleB dataset is shown in Fig. 6. The x-axis denotes the number of iterations, while the y-axis represents the errors defined in Eq. (24). We can see that, after several iterations, the errors witness a quick drop until a stable value. In all experiments, the proposed JLMVC can reach the smallest residual within 50 iterations. To further investigate the empirical convergence of JLMVC, Fig. 6 also reports the ACC and NMI values with respect to iterations on Extended YaleB dataset. Before the first 10 iterations, JLMVC does not reach a meaningful accuracy. But after that, JLMVC achieves promising ACC and NMI values higher than those of all competing methods except Kt-SVD-MSC. This shows that the proposed JLMVC is an excellent multi-view clustering method.

(3) The effect of $\mathcal{Z}$ and S: The proposed JLMVC achieves the joint learning of the representation tensor $\mathcal{Z}$ and affinity matrix S. However, most existing MVC methods follow two separate steps to construct $\mathcal{Z}$ and S. To investigate the effect of $\mathcal{Z}$ and S, we perform a test by setting $\lambda = 0$. In this test, we simply obtain $\mathcal{Z}$ and then construct $S = \frac{1}{V}\sum_v \left(|Z^{(v)}| + |Z^{(v)^T}|\right)$. This simple variant of JLMVC is denoted as JLMVC-$\mathcal{Z}$. Table V reports clustering results of JLMVC and JLMVC-$\mathcal{Z}$. It is easy to see that JLMVC achieves superior clustering results over JLMVC-$\mathcal{Z}$ in all cases. The average improvement of JLMVC is around 17.06% and 16.23% over JLMVC-$\mathcal{Z}$ with respect to ACC and NMI, respectively, indicating that construction of $\mathcal{Z}$ and

TABLE V
PERFORMANCE (ACC/NMI) OF JLMVC AND ITS VARIANTS ON DIFFERENT DATASETS.

	ACC/NMI								
	YaleB	Yale	ORL	Prokaryotic	Wikipedia	COIL-20	CMU-PIE	Scene-15	Average
JLMVC	0.910/0.897	1.000/1.000	0.983/0.996	0.766/0.487	0.587/0.552	0.945/0.970	0.918/0.970	0.988/0.975	0.8871/0.8559
JLMVC-$\mathcal{Z}$	0.532/0.504	0.386/0.485	0.977/0.995	0.604/0.330	0.544/0.446	0.919/0.955	0.786/0.867	0.984/0.967	0.7165/0.6936
JLMVC-nk	0.534/0.468	0.932/0.951	0.966/0.993	0.722/0.379	0.584/0.521	0.894/0.935	0.891/0.955	0.978/0.963	0.8126/0.7706

TABLE VI
AVERAGE RUNNING TIME (IN SECONDS) ON ALL DATABASES.

Data	MLAP	DiMSC	LT-MSC	MLAN	t-SVD-MSC	HLR-M^{2}VS	Kt-SVD-MSC	JLMVC
Yale	35.06	2.85	17.98	0.89	8.24	5.83	24.74	8.96
YaleB	250.42	30.54	128.15	4.85	54.19	54.54	371.92	68.57
ORL	128.32	13.16	65.28	1.60	34.85	23.66	22.55	19.68
Wikipedia	75.49	22.74	28.67	3.35	5.46	14.18	347.14	40.53
Prokaryotic	65.51	17.65	29.34	2.89	7.32	16.02	35.72	32.18
COIL-20	1826.51	617.29	874.91	31.03	169.10	314.16	344.36	322.98
CMU-PIE	26683.28	24281.78	14645.16	6272.19	6344.51	10677.92	32986.18	13825.17
Scene-15	13825.53	12449.36	7705.87	3318.62	3429.46	6274.69	16915.99	7680.65

S simultaneous can boost the clustering performance.

(4) Ablation study on the kernel trick: To investigate the effect of the kernel trick, we also carry out the model in Eq. (3), denoted as JLMVC-nk. Like JLMVC, JLMVC-nk also learns the representation tensor and affinity matrix simultaneously without the kernel trick. This means that the affinity matrix is constructed from the the original multimedia data (usually nonlinear separable). The ACC and NMI values of JLMVC-nk are reported in the last row of Table V. One can see that JLMVC achieves better clustering results than JLMVC-nk in all cases. A typical example is the Extended YaleB dataset whose multiple features are diverse as shown in Fig. 4. This indicates that the kernel trick can handle the nonlinearity and boost the multi-view clustering performance.

(5) Runtime: Since the computation time of a method is also an evaluation factor, we give a runtime comparison of the proposed JLMVC and several competitors. Table VI reports the runtime comparison results. All experiments are implemented in Matlab 2016a on a workstation with 3.50GHz CPU and 16GB RAM. From Table VI, the methods with the average time from low to high are MLAN, t-SVD-MSC, HLR-M^{2}VS, JLMVC, LT-MSC, DiMSC, MLAP, and Kt-SVD-MSC. MLAN costs the shortest processing time and the proposed JLMVC belongs to the middle-ranking group. All methods except for MLAN should compute the singular value decomposition and matrix inversion. This leads to a high computation cost. Although MLAN is the most efficient one, it has an unstable performance. The reason is that MLAN uses the raw data to learn the similarity matrix and the raw data are easily contaminated by noise. Other methods impose the low-rank constraint on the representation matrix (or tensor) and use the sparse regularizer to remove noise. They can construct a reliable similarity matrix.

V. CONCLUSIONS

In this paper, we proposed a novel method called JLMVC to solve the multi-view clustering problem, based on the low-rank tensor representation and "kernel trick". In JLMVC, instead of capturing a low-rank representation matrix among all views, the tensor singular value decomposition-based tensor nuclear norm was used to learn the representation tensor so as to explore the high order correlations among different views. Using the kernel trick, the original multimedia data was implicitly mapped from the input data space into a new feature space to overcome the difficulty of nonlinearity in real applications. To make full use of the high correlation between the representation tensor and affinity matrix, the proposed JLMVC achieved the joint learning of the representation tensor and affinity matrix. Thus, the learned affinity matrix has the potential to boost the clustering performance which was demonstrated by extensive experiments on eight multimedia datasets. Our future work will design a fast and efficient multi-view clustering method. One possible solution is using the Frank-Wolfe algorithm to reduce the computation complexity of the singular value decomposition.

VI. ACKNOWLEDGMENT

The authors would like to thank the editors and the anonymous reviewers for their constructive comments which helped to improve the quality of this article. The authors wish to gratefully acknowledge Prof. Changqing Zhang from Tianjin University and Prof. Yuan Xie from East China Normal University for sharing multi-view datasets and codes.

REFERENCES

[1] S. Yang, L. Li, S. Wang, W. Zhang, Q. Huang, and Q. Tian, "SkeletonNet: A hybrid network with a skeleton-embedding process for multi-view image representation learning," *IEEE Trans. Multimedia*, 2019.

[2] Z. Zhang, Y. Xie, W. Zhang, and Q. Tian, "Effective image retrieval via multilinear multi-index fusion," *IEEE Trans. Multimedia*, 2019.

[3] S. K. Kuanar, K. B. Ranga, and A. S. Chowdhury, "Multi-view video summarization using bipartite matching constrained optimum-path forest clustering," *IEEE Trans. Multimedia*, vol. 17, no. 8, pp. 1166–1173, 2015.

[4] X. Wu, C.-W. Ngo, and A. G. Hauptmann, "Multimodal news story clustering with pairwise visual near-duplicate constraint," *IEEE Trans. Multimedia*, vol. 10, no. 2, pp. 188–199, 2008.

[5] C. Tang, X. Zhu, X. Liu, M. Li, P. Wang, C. Zhang, and L. Wang, "Learning joint affinity graph for multi-view subspace clustering," *IEEE Trans. Multimedia*, 2018.

[6] C. Zhang, H. Fu, S. Liu, G. Liu, and X. Cao, "Low-rank tensor constrained multiview subspace clustering," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1582–1590.

[7] 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, 2018.

[8] 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.*, 2018.

[9] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. Neural Inf. Process. Syst.*, 2002, pp. 849–856.

[10] X. Guo, "Robust subspace segmentation by simultaneously learning data representations and their affinity matrix," in *Proc. Joint Conf. Artif. Intell.*, 2015.

[11] X. Peng, Z. Yu, Z. Yi, and H. Tang, "Constructing the l2-graph for robust subspace learning and subspace clustering," *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 1053–1066, 2017.

[12] F. Nie, X. Wang, and H. Huang, "Clustering and projected clustering with adaptive neighbors," in *Proc. ACM SIGKDD*, 2014, pp. 977–986.

[13] F. Nie, H. Wang, C. Deng, X. Gao, X. Li, and H. Huang, "New l_1-norm relaxations and optimizations for graph clustering," in *Proc. AAAI Conf. Artif. Intell.*, 2016.

[14] F. Nie, X. Wang, M. I. Jordan, and H. Huang, "The constrained laplacian rank algorithm for graph-based clustering," in *Proc. AAAI Conf. Artif. Intell.*, 2016.

[15] K. Zhan, C. Zhang, J. Guan, and J. Wang, "Graph learning for multiview clustering," *IEEE Trans. Cybern.*, no. 99, pp. 1–9, 2017.

[16] F. Nie, G. Cai, J. Li, and X. Li, "Auto-weighted multi-view learning for image clustering and semi-supervised classification," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1501–1511, 2018.

[17] F. Nie, J. Li, X. Li *et al.*, "Self-weighted multiview clustering with multiple graphs." in *Proc. Joint Conf. Artif. Intell.*, 2017, pp. 2564–2570.

[18] H. Wang, Y. Yang, and B. Liu, "GMC: Graph-based multi-view clustering," *IEEE Trans. Knowl. Data Eng.*, 2019.

[19] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171–184, 2013.

[20] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2765–2781, 2013.

[21] C. Lu, J. Feng, Z. Lin, T. Mei, and S. Yan, "Subspace clustering by block diagonal representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 2, pp. 487–501, 2019.

[22] Y. Wang, L. Wu, X. Lin, and J. Gao, "Multiview spectral clustering via structured low-rank matrix factorization," *IEEE Trans. Neural Netw. Learn. Syst.*, no. 99, pp. 1–11, 2018.

[23] X. Wang, X. Guo, Z. Lei, C. Zhang, and S. Z. Li, "Exclusivity-consistency regularized multi-view subspace clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 923–931.

[24] Z. Kang, H. Pan, S. C. H. Hoi, and Z. Xu, "Robust graph learning from noisy data," *IEEE Trans. Cybern.*, pp. 1–11, 2019.

[25] M. Yin, S. Xie, Z. Wu, Y. Zhang, and J. Gao, "Subspace clustering via learning an adaptive low-rank graph," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3716–3728, 2018.

[26] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Comput.*, vol. 15, no. 6, pp. 1373–1396, 2003.

[27] S. Yi, Y. Liang, Z. He, Y. Li, W. Liu, and Y.-m. Cheung, "Dual pursuit for subspace learning," *IEEE Trans. Multimedia*, 2018.

[28] P. Ji, T. Zhang, H. Li, M. Salzmann, and I. Reid, "Deep subspace clustering networks," in *Proc. Neural Inf. Process. Syst.*, 2017, pp. 24–33.

[29] H. Zhao, Z. Ding, and Y. Fu, "Multi-view clustering via deep matrix factorization," in *Proc. AAAI Conf. Artif. Intell.*, 2017.

[30] Z. Zhang, L. Liu, J. Qin, F. Zhu, F. Shen, Y. Xu, L. Shao, and H. Tao Shen, "Highly-economized multi-view binary compression for scalable image clustering," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 717–732.

[31] G. Chao, J. Sun, J. Lu, A.-L. Wang, D. D. Langleben, C.-S. Li, and J. Bi, "Multi-view cluster analysis with incomplete data to understand treatment effects," *Inf. Sci.*, vol. 494, pp. 278–293, 2019.

[32] G. Chao, S. Sun, and J. Bi, "A survey on multi-view clustering," *arXiv preprint arXiv:1712.06246*, 2017.

[33] B. Cheng, G. Liu, J. Wang, Z. Huang, and S. Yan, "Multi-task low-rank affinity pursuit for image segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.* IEEE, 2011, pp. 2439–2446.

[34] R. Xia, Y. Pan, L. Du, and J. Yin, "Robust multi-view spectral clustering via low-rank and sparse decomposition," in *Proc. AAAI Conf. Artif. Intell.*, 2014.

[35] C. Zhang, Q. Hu, H. Fu, P. Zhu, and X. Cao, "Latent multi-view subspace clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4279–4287.

[36] S. Luo, C. Zhang, W. Zhang, and X. Cao, "Consistent and specific multi-view subspace clustering," in *Proc. AAAI Conf. Artif. Intell.*, 2018.

[37] C. Zhang, H. Fu, Q. Hu, X. Cao, Y. Xie, D. Tao, and D. Xu, "Generalized latent multi-view subspace clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2018.

[38] J. Liu, C. Wang, J. Gao, and J. Han, "Multi-view clustering via joint nonnegative matrix factorization," in *Proc. SIAM Int. Conf. Data Min.* SIAM, 2013, pp. 252–260.
[39] M. Yin, J. Gao, S. Xie, and Y. Guo, "Multiview subspace clustering via tensorial t-product representation," *IEEE Trans. Neural Netw. Learn. Syst.*, no. 99, pp. 1–14, 2018.
[40] V. M. Patel and R. Vidal, "Kernel sparse subspace clustering," in *Proc. IEEE Int. Conf. Image Process.*, 2014, pp. 2849–2853.
[41] S. Xiao, M. Tan, D. Xu, and Z. Y. Dong, "Robust kernel low-rank representation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 11, pp. 2268–2281, 2016.
[42] Y. Chen, X. Xiao, and Y. Zhou, "Multi-view clustering via simultaneously learning graph regularized low-rank tensor representation and affinity matrix," in *Proc. Int. Conf. Multimedia Expo.* IEEE, 2019, pp. 1348–1353.
[43] Y. Qu, J. Liu, Y. Xie, and W. Zhang, "Robust kernelized multi-view self-representations for clustering by tensor multi-rank minimization," *arXiv preprint arXiv:1709.05083*, 2017.
[44] M. E. Kilmer and C. D. Martin, "Factorization strategies for third-order tensors," *Linear Algebra Appl.*, vol. 435, no. 3, pp. 641–658, 2011.
[45] Y. Chen, Y. Guo, Y. Wang, D. Wang, C. Peng, and G. He, "Denoising of hyperspectral images using nonconvex low rank matrix approximation," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 9, pp. 5366–5380, 2017.
[46] Y. Chen, X. Xiao, and Y. Zhou, "Low-rank quaternion approximation for color image processing," *IEEE Trans. Image Process.*, 2019.
[47] Y. Chen, S. Wang, and Y. Zhou, "Tensor nuclear norm-based low-rank approximation with total variation regularization," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 6, pp. 1364–1377, 2018.
[48] W. Hu, D. Tao, W. Zhang, Y. Xie, and Y. Yang, "The twist tensor nuclear norm for video completion," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 12, pp. 2961–2973, 2017.
[49] Y. Chen, Y. Wang, M. Li, and G. He, "Augmented lagrangian alternating direction method for low-rank minimization via non-convex approximation," *Signal, Image Video Process.*, vol. 11, no. 7, pp. 1271–1278, 2017.
[50] T. Zhou, C. Zhang, C. Gong, H. Bhaskar, and J. Yang, "Multiview latent space learning with feature redundancy minimization," *IEEE Trans. Cybern.*, 2018.
[51] L. Fei-Fei and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 2. IEEE, 2005, pp. 524–531.
[52] X. Cao, C. Zhang, H. Fu, S. Liu, and H. Zhang, "Diversity-induced multi-view subspace clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 586–594.
[53] F. Nie, L. Tian, and X. Li, "Multiview clustering via adaptively weighted procrustes," in *Proc. ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.* ACM, 2018, pp. 2022–2030.

Yongyong Chen received his B.S. and M.S. degrees in the College of Mathematics and Systems Science, Shandong University of Science and Technology, Qingdao, China and visited the National Key Lab for Novel Software Technology in Nanjing University as an exchange student in 2017. He is now pursuing his Ph.D. degree in the Department of Computer and Information Science, University of Macau, Macau, China. His research interests include (non-convex) low-rank and sparse matrix/tensor decomposition models, with applications to image processing, data mining and computer vision.

Xiaolin Xiao received the B.E. degree from Wuhan University, China, in 2013 and the Ph.D. degree from University of Macau, Macau, China, in 2019. Currently, she is a Postdoc Fellow with the School of Computer Science and Engineering, South China University of Technology, China. Her research interests include superpixel segmentation, saliency detection, and color image processing and understanding.

Yicong Zhou (M'07-SM'14) received his B.S. degree from Hunan University, Changsha, China, and his M.S. and Ph.D. degrees from Tufts University, Massachusetts, USA, all in electrical engineering.

He is an Associate Professor and Director of the Vision and Image Processing Laboratory in the Department of Computer and Information Science at University of Macau. His research interests include image processing and understanding, computer vision, machine learning, and multimedia security.

Dr. Zhou is a senior member of the International Society for Optical Engineering (SPIE). He was a recipient of the Third Price of Macau Natural Science Award in 2014. He is a Co-Chair of Technical Committee on Cognitive Computing in the IEEE Systems, Man, and Cybernetics Society. He serves as an Associate Editor for *IEEE Transactions on Circuits and Systems for Video Technology*, *IEEE Transactions on Geoscience and Remote Sensing*, and four other journals.