






# Self-Paced Enhanced Low-Rank Tensor Kernelized Multi-View Subspace Clustering

Yongyong Chen , Shuqin Wang, Xiaolin Xiao , Youfa Liu , Zhongyun Hua , *Member, IEEE*,  
and Yicong Zhou , *Senior Member, IEEE*

**Abstract**—This paper addresses the multi-view subspace clustering problem and proposes the self-paced enhanced low-rank tensor kernelized multi-view subspace clustering (SETKMC) method, which is based on two motivations: (1) singular values of the representations and multiple instances should be treated differently. The reasons are that larger singular values of the representations usually quantify the major information and should be less penalized; samples with different degrees of noise may have various reliability for clustering. (2) many existing methods may cause the degraded performance when multi-view features reside in different nonlinear subspaces. This is because they usually assumed that multiple features lie within the union of several linear subspaces. SETKMC integrates the nonconvex tensor norm, self-paced learning, and kernel trick into a unified model for multi-view subspace clustering. The nonconvex tensor norm imposes different weights on different singular values. The self-paced learning gradually involves instances from more reliable to less reliable ones while the kernel trick aims to handle the multi-view data in nonlinear subspaces. One iterative algorithm is proposed based on the alternating direction method of multipliers. Extensive results on seven real-world datasets show the effectiveness of the proposed SETKMC compared to fifteen state-of-the-art multi-view clustering methods.

**Index Terms**—Multi-view clustering, low-rank tensor representation, kernel, enhanced low-rank representation, self-paced learning.

Manuscript received 9 May 2021; revised 15 August 2021; accepted 7 September 2021. Date of publication 22 September 2021; date of current version 9 August 2022. This work was supported in part by the National Natural Science Foundation of China under Grants 62106063, 62106081, and 62071142, in part by Shenzhen College Stability Support Plan under Grants GXWD20201230155427003-20200824113231001 and GXWD20201230155427003-20200824210638001, in part by the Fundamental Research Funds for the Central Universities under Grant 2662020XXQD002, in part by Cultivation Project under Grant 2662021JC008, and in part by the University of Macau (File no. MYRG2018-00136-FST). (*Corresponding authors: Youfa Liu; Zhongyun Hua.*)

Yongyong Chen and Zhongyun Hua are with the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen 518055, China, and with the Bio-Computing Research Center, Harbin Institute of Technology, Shenzhen 518055, China, and also with the Shenzhen Key Laboratory of Visual Object Detection and Recognition, Shenzhen 518055, China (e-mail: YongyongChen.cn@gmail.com, huazyum@gmail.com).

Shuqin Wang is with the Institute of Information Science, Beijing Jiaotong University, Beijing 100044, China (e-mail: ShuqinWang.cn@hotmail.com).

Xiaolin Xiao is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China (e-mail: shellyxiaolin@gmail.com).

Youfa Liu is with the College of Informatics, Huazhong Agricultural University, Wuhan 430070, China (e-mail: liuyoufa@mail.hzau.edu.cn).

Yicong Zhou is with the Department of Computer and Information Science, University of Macau, Macau 999078, China (e-mail: yicongzhou@um.edu.mo).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TMM.2021.3112230>.

Digital Object Identifier 10.1109/TMM.2021.3112230

## I. INTRODUCTION

A VERITABLE multi-view data deluge has been unleashed by the advance of information technology. For example, beyond the original image, several different types of features of multimedia data including LBP, Gabor, SIFT, and powerful deep features are extracted for face recognition, action recognition [1], [2]. The same content can be recorded with multimedia data such as images, videos, audio, and documents, etc [3]. Other examples include multimedia retrieval and video surveillance, among which multiple features or videos are extracted. In multimedia retrieval, images and videos can be described by some typical multimedia features such as bag-of-visual-words (BOW), Fisher vectors, Vector of locally aggregated descriptors (VLAD), and deep features. In multi-camera video surveillance, human activities of interest are recorded by multi-cameras, where each camera corresponds to a view. Therefore, multimedia data are closely related with multi-view data [4]. These multimedia data are captured from diverse viewpoints or various sources and consequently result in complex characteristics including consistent and diverse information. Considering the difficulty of labeled samples and multi-view features of multimedia data sampled from a union of subspaces, multi-view subspace clustering (MVSC) [2], [5]–[7], partitioning a large number of unlabeled multimedia data into several distinct clusters, has recently flourished in motion segmentation, person re-identification, background subtraction, video image recognition, social multimedia data clustering, gene detection and so on [8].

Generally, both MVSC and single-view subspace clustering (SVSC) methods follow the same pipeline: representation learning and spectral clustering [9]. The first step aims to find a representation of multimedia data by pursuing a desirable affinity matrix with some specific characteristics such as low-rankness [10]–[12], sparsity [13], [14] and block diagonality [15], while the last step aims to obtain the clustering results by inputting the affinity matrix into the spectral clustering algorithm [16]. The representation learning (*i.e.*, the first step) can be roughly formulated as

$$\min_{Z^v} \sum_{v=1}^V \frac{1}{2} \|X^v - X^v Z^v\|_l + \lambda \mathcal{R}(Z^v) \quad (1)$$

where  $X^v$  and  $Z^v$  denote the  $v$ -th multi-view feature and its corresponding representation, respectively;  $\|\cdot\|_l$  aims to measure the noises, such as the Gaussian noise ( $l_2$ -norm), impulse noise ( $l_1$ -norm), sample-specific noise ( $l_{2,1}$ -norm);  $\mathcal{R}(\cdot)$  denotes the

regularizer. For example, Liu *et al.* [10] proposed the low-rank representation (LRR) method for SVSC. Zhang *et al.* [1] and Xie *et al.* [17] extended LRR from the matrix form into the tensor form to exploit the low-rank tensor property of the representation tensor in the vector space and the tensor space, respectively. Chen *et al.* [2] exploited the Tucker decomposition to encode the low-rank property. Beyond the above self-representation, the studies in [11], [18], [19] adopted the Markov chain to construct the multi-view transition probability matrices which are decomposed into the sum of a low-rank term and a sparse component. Despite the great success of subspace clustering methods, three challenging problems may arise. (1) using the convex  $l_1$ -norm (or nuclear norm) to characterize the sparsity (or low-rankness) may amplify the approximation error of the original nonconvex  $l_0$ -norm and rank function. Many studies [20]–[22] have pointed out that both of them are not accurate approximations of the original nonconvex  $l_0$ -norm and rank function. (2) since most of them assume that all multimedia data lie within the union of several linear subspaces, their clustering performance may not be guaranteed when multiple views come from the non-linear subspaces [23]. (3) different norms such as  $l_2$ -norm,  $l_1$ -norm,  $l_{2,1}$ -norm, and mixture of Gaussians, restrict the sample noise but still treat them equally, and thus it is impossible to avoid the disturbance of abnormal samples to clustering.

To solve the first challenge, existing solutions designed an effective alternative closer to the rank function. For example, the study in [22] uses three non-convex functions instead of the nuclear norm in denoising to achieve powerful results. To address the second challenge, two strategies *i.e.*, manifold learning and kernel trick were adopted to represent nonlinearity of multimedia data. The representative method of manifold learning is the Laplacian regularizer. Specifically, Yin *et al.* [24] incorporated the manifold learning with LRR to capture the global low-rankness and local geometrical structure embedding with nonlinearity for SVSC. Similar ideas were followed in [25], [26] for MVSC. But the Laplacian regularizer considers only the local structure of multimedia data through the similarity of points to points. Some state-of-the-art methods using the kernel trick include [23], [27], [28]. For example, Xiao *et al.* [23] applied kernel trick to map the original feature from the original input space into a new feature space, such that the mapped features may reside in multiple linear subspaces. Inspired by the block diagonal representation [15] and kernel trick, Xie *et al.* [27] proposed the implicit block diagonal LRR. Since different samples may contain different degrees of noise, it is more reasonable to treat them separately. One possible solution to handle the third challenge is the self-paced learning [29], which processes the data sample from the simple order to complex order. In summary, most of existing literatures merely consider to solve one or two of the above three challenges. To the best of our knowledge, there is no work to address these three challenges simultaneously.

In this paper, we propose the Self-paced Enhanced low-rank Tensor Kernelized Multi-view subspace Clustering (SETKMC) method to simultaneously solve the three challenges described above. Instead of the convex  $l_1$ -norm and nuclear norm, SETKMC proposes one novel nonconvex tensor norm by exploiting one nonconvex function to less penalize the large tensor

singular values, subsequently yielding a nonconvex and much challenging model. SETKMC uses the kernel trick to handle the multi-view data in the nonlinear subspaces. In order to distinguish samples with different reliability, SETKMC also adopts the self-paced learning to incrementally process instances from the simple order to complex order. Thus, the proposed SETKMC is a unified model that simultaneously solves the tensor nuclear norm's biased approximation of tensor rank, the nonlinearity and the side effect caused by some difficult instances. Based on the alternating direction method of multipliers, we propose an effective algorithm to solve the nonconvex SETKMC model, in which the low-rank tensor term is derived by the difference of convex method. The contributions of this paper are summarized as follows:

- We propose a novel method, namely self-paced enhanced low-rank tensor kernelized multi-view subspace clustering method (SETKMC), which unifies the nonconvex tensor norm, the kernel trick, and the self-paced learning.
- Unlike most existing methods which over-penalize the larger singular values and ignore the side effect caused by some difficult instances, SETKMC proposes the enhanced low-rank tensor norm to estimate the singular values more accurately than the tensor nuclear norm and adopts the self-paced learning strategy to process from easy example to complex example for clustering. Besides, SETKMC utilizes the kernel trick to overcome the nonlinearity.
- Extensive experiments on seven real-world databases demonstrate the effectiveness of the proposed SETKMC compared to several state-of-the-art convex, kernel and deep multi-view clustering methods.

The remainder of this paper is organized as follows. The related works for multi-view clustering are summarized in Section II. Some preliminaries are shown in Section III. Section IV presents the proposed SETKMC model and designs an effective algorithm to solve the SETKMC model. Section V reports the results of extensive experiments and model analysis. The conclusion of this paper is summarized in Section VI.

## II. RELATED WORK

In this section, we mainly review multi-view tensor clustering methods, multi-view kernel clustering ones and self-paced learning.

### A. Multi-View Tensor Clustering

Recent advances have proven that multi-view tensor clustering methods have achieved superior performance over multi-view matrix ones. This is because they stack the representation matrices from multiple views as a 3-dimensional tensor to capture two-dimensional spatial correlation and one-dimensional view correlation. The seminal works are [1] and [17], both of which followed the general model

$$\min_{\mathcal{Z}, E} \mathcal{R}(\mathcal{Z}) + \alpha \sum_{v=1}^V \|E^v\|_{2,1} \quad \text{s.t. } X^v = X^v \mathcal{Z}^v + E^v, \mathcal{Z} = \Phi(\mathcal{Z}^1, \mathcal{Z}^2, \dots, \mathcal{Z}^V), \quad (2)$$

where  $V$  is the number of all views.  $V$  features  $\{X^v \in \mathbb{R}^{d_v \times n}\}_{v=1}^V$  are to measure the same object. Matrix  $X^v$  ( $v = 1, 2, \dots, V$ ) denotes the  $v$ -th feature matrix.  $Z^v \in \mathbb{R}^{n \times n}$  is the corresponding representation matrix.  $d_v$  is the dimension of a sample vector in the  $v$ -th feature matrix.  $n$  is the total number of data points.  $\mathcal{R}(\mathcal{Z})$  is the regularizer to measure the low-rankness of the representation tensor  $\mathcal{Z}$ . For example, the study in [1] used the unfolding-based tensor nuclear norm, while that in [17] used the tensor nuclear norm in Eq. (3). Chen *et al.* [30] used the Tucker decomposition to capture the low-rankness. Under the low-rank tensor representation framework, Wu *et al.* [31] used the projected graph learning to learn the view-specific affinity matrix, eliminating the curse of high-dimensional data. Xu *et al.* [32] imposed the low-rank tensor property on a third-order tensor, in which each frontal slice is the indicator matrix of the  $v$ -th view. Other followers include [18], [19], [33], [34]. However, the tensor norm in these methods treats the different singular values equally, leading to over-penalizing singular values and the biased estimation.

### B. Multi-View Kernel Clustering

Existing multi-view kernel clustering models can be roughly classified into two categories based on how the kernels are used. The first category is to use the predefined kernels, such as Linear kernel, Polynomial kernel, and Gaussian kernel, and then combine these kernels either linearly or nonlinearly [35]. For example, reference [36] proposed the kernel-based weighted multi-view clustering method, in which different weights are imposed on several given kernel matrices according to the quality of view information. Yu *et al.* [37] integrated the kernel trick with the traditional  $k$ -means algorithm. Huang *et al.* [38] developed the auto-weighted multi-view clustering method via kernelized graph learning, in which similarity relationships were learned in kernel spaces. The second category is to map non-linear data into a new feature space, in which the multiple nonlinear features may reside in several linear subspaces. For instance, in the single view scenario, both of [23] and [27] adopted the kernel-induced mapping to handle the nonlinear data and then learned the representation from the new feature space. Inherited from the above idea, Xie *et al.* [28] developed a kernelized version of [17] to capture multiple views correlation. Chen *et al.* [9] used not only the kernel trick, but also the joint optimization to jointly learn the kernel representation tensor and affinity matrix.

### C. Self-Paced Learning

Self-paced learning [39] and curriculum learning [40] are inspired by the learning process of humans/animals, *i.e.*, learning the model iteratively from easy samples to complex ones in a self-paced fashion. Due to the advance of avoiding bad local minima and promising performance, self-paced learning has been widely explored in several fields. For example, Zhao *et al.* [41] integrated the self-paced learning with matrix factorization to handle the structure from motion and background subtraction. Zhang *et al.* [42] proposed a self-paced multiple-instance learning framework for co-saliency detection. Zhou *et al.* [43] integrated the self-paced learning and ensemble learning into a

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 $\mathcal{X}$
$\hat{\mathcal{X}} = \text{fft}(\mathcal{X}, [], 3)$	fast Fourier transformation along tube fiber
$n, V, d_v$	the number of samples, views, feature dimension
$X^v, E^{(v)} \in \mathbb{R}^{d_v \times n}$	feature matrix and error of the $v$ -th view
$\pi, \omega^v$	map function, weight of the $v$ -th view
$\Phi(Z^1, Z^2, \dots, Z^V)$	operator to form tensor $\mathcal{Z} \in \mathbb{R}^{n \times n \times V}$
$\{\mathbf{K}^v\}_{v=1}^V, f(\omega, \beta)$	kernel matrices, regularizer for self-paced learning
$\mathcal{Y}, \Pi \in \mathbb{R}^{n \times n \times V}$	auxiliary variable, Lagrange multiplier
$e^{(x)}$	exponential function
$\gamma, \lambda, \beta, \rho$	parameters
$\ \cdot\ _{2,1}, \ \cdot\ _F$	$\ell_{2,1}$ -norm, Frobenius norm
$\ \cdot\ _{\oplus}, \ \cdot\ _{\infty}$	t-SVD-nuclear norm, infinity norm

unified framework for ensemble clustering. Xu *et al.* [44] explored the self-paced learning for multi-view clustering.

Besides the aforementioned methods, there are also some other related works. For example, the study in [45] developed the enhanced tensor robust principal component analysis model for image recovery. Zhang *et al.* [46] solved the MVSC in the latent space. Due to the outstanding representation capacity, Wang *et al.* [47] proposed a deep MVSC method by unified and discriminative learning. Similarly, Xia *et al.* [48] developed the multi-view self-supervised graph convolutional clustering network. Considering its comprehensive representation, multi-view learning has recently been received substantial attentions in different applications including large-scale multimedia search [49], [50], large-scale effective ensemble adversarial attacks [51], 3-D object retrieval and classification [52].

## III. PRELIMINARIES

We aim to develop an enhanced low-rank tensor representation to well describe the low-rankness of the representation tensor. Thus, we start with some definitions that will be used to derive the enhanced low-rank tensor norm as defined in Eq. (4). Some notations are summarized in Table I. For a tensor  $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ , its block circular matrix  $\text{bcirc}(\mathcal{X})$  and block diagonal matrix  $\text{bdiag}(\mathcal{X})$  are defined as

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

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

The block vectorization is defined as  $\text{bvec}(\mathcal{X}) = [\mathcal{X}^{(1)}; \dots; \mathcal{X}^{(n_3)}]$ . The inverse operations of  $\text{bvec}$  and  $\text{bdiag}$  are defined as  $\text{bvfold}(\text{bvec}(\mathcal{X})) = \mathcal{X}$  and  $\text{bdfold}(\text{bdiag}(\mathcal{X})) = \mathcal{X}$ , 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,  $\mathcal{X} * \mathcal{Y} = \text{bvfold}(\text{bcirc}(\mathcal{X}) * \text{bvec}(\mathcal{Y}))$ . 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  $\mathcal{X}^T * \mathcal{X} = \mathcal{X} * \mathcal{X}^T = \mathcal{I}$ .

**Definition 3.1: (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.

The main purpose of t-SVD is to make the tensor decomposition similar to that of the matrix singular value decomposition. After that, the tensor multirank and its convex surrogate, *i.e.*, tensor nuclear norm can be defined as follows:

**Definition 3.2: (Tensor multirank)** The multirank of a tensor  $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$  is a vector whose  $i$ -th element is the rank of the  $i$ -th frontal slice of  $\hat{\mathcal{X}}$ .

**Definition 3.3: (Tensor nuclear norm)** The tensor nuclear norm  $\|\mathcal{Z}\|_{\oplus}$  of a tensor  $\mathcal{Z} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$  is defined as the average of the singular values of all the frontal slices of  $\hat{\mathcal{Z}}$ , *i.e.*,

$$\|\mathcal{Z}\|_{\oplus} = \frac{1}{n_3} \sum_{k=1}^{n_3} \|\hat{\mathcal{Z}}^{(k)}\|_*. \quad (3)$$

The above tensor nuclear norm has been widely in [17], [19], [28] for MVSC.

#### IV. THE PROPOSED SETKMC MODEL

In this section, we propose the SETKMC model, and then solve SETKMC using the alternating direction method of multipliers (ADMM) and the difference of convex method.

##### A. Proposed SETKMC

Since the existing MVSC methods are introduced to handle data points drawn from multiple linear subspaces, they may not produce competitive performance when data points are from nonlinear subspaces. To overcome this limitation, the studies in [23], [28] adopted the kernel trick to deal with the single-view clustering and multi-view clustering tasks, respectively. Reference [23] seeks a low-rank matrix representation using the matrix nuclear norm while Reference [28] seeks a low-rank tensor representation using the tensor nuclear norm defined in Eq. (3). However, both of them ignore the fact that the matrix and tensor nuclear norms are proved to overestimate the nonzero singular values since they impose the equal weights on the larger and smaller singular values. Besides, both of them fail to consider the differences among samples due to noise and outliers. Motivated by the promising performance of the recently proposed nonconvex penalty functions [20], [21], [53], we propose a novel MVSC method using the self-paced enhanced low-rank tensor kernelized representation. The enhanced low-rank tensor norm (ELTN) is defined as

$$\|\mathcal{Z}\|_{ELTN} = \sum_j \sum_i (1 - e^{-\frac{\sigma_i(\hat{\mathcal{Z}}^{(j)})}{\gamma}}), \quad (4)$$

where  $e^{(x)}$  is the exponential function,  $\sigma_i(\hat{\mathcal{Z}}^{(j)})$  is the  $i$ -th singular value of the  $j$ -th frontal slice of  $\hat{\mathcal{Z}}$ ,  $\gamma > 0$  is a constant. According to Eq. (3), one can see that the equal weights are imposed on all singular values, regardless of their values. However, our ELTN can impose different weights on different singular values using the exponential function. Thus, the main idea of ELTN is to estimate the singular values more accurately than the tensor nuclear norm in Eq. (3).

To overcome the nonlinearity of multi-view features, we follow the kernel theory [26] to map the original data  $\mathbb{R}^d$  into a high-dimensional kernel Hilbert space  $\mathbb{H}^k$  (usually implicitly defined). Consequently, each view  $X^v$  is represented by a kernel matrix  $K^v \in \mathbb{R}^{n \times n}$ , and the  $(i, j)$ -th element of  $K^v$  is computed by  $K_{i,j}^v = \mathcal{K}(x_i^v, x_j^v) = \pi(x_i^v)^T \pi(x_j^v)$ , where  $\mathcal{K}(x_i^v, x_j^v)$  denotes the kernel function,  $\pi$  is a map function. Based on the above discussions, the traditional self-representation  $X^v = X^v Z^v + E^v$  would be formulated as  $\pi(X^v) = \pi(X^v) Z^v + E^v$ , assuming  $\pi(X^v)$  resides in multiple linear subspaces. Besides, since each sample may contain the sample-specific error, [10] used the  $l_{2,1}$ -norm to eliminate noises. Thus, the self-representation can be further expressed as

$$\|\pi(X^v) - \pi(X^v) Z^v\|_{2,1} = \sum_{i=1}^n \left( P_i^{vT} K^v P_i^v \right)^{\frac{1}{2}}, \quad (5)$$

where  $P^v = I - Z^v$ . Besides, most existing methods use all samples for multi-view clustering without considering their different credibility [54]. Intuitively, compared with easier reliable samples, ones infected with noise and outliers should be gradually added to the learning process. To achieve this, we adopt the self-paced learning theory to learn easy to hard samples. Accordingly, the proposed SETKMC is formulated as

$$\left\{ \begin{array}{l} \min_{\mathcal{Z}, P, \omega} \|\mathcal{Z}\|_{ELTN} + \lambda \sum_{v=1}^V \sum_{i=1}^n \omega_i^v g^v(P_i^v) + f(\omega, \beta) \\ s.t. \quad P^v = I - Z^v, \quad v = 1, 2, \dots, V, \\ \quad \mathcal{Z} = \Phi(Z^1, Z^2, \dots, Z^V), \\ \quad \omega^v = [\omega_1^v, \omega_2^v, \dots, \omega_n^v] \in [0, 1]^n, \end{array} \right\} \quad (6)$$

where  $g^v(P^v) = \sum_{i=1}^n (p_i^{vT} K^v p_i^v)^{1/2}$  is derived from Eq. (5).  $\omega^v$  is composed of the weights of  $n$  samples in the  $v$ -th view. Parameter  $\beta > 0$  is to control the speed of samples.

To well understand the proposed SETKMC model in Eq. (6), several remarks are shown as follows:

- Without considering the self-paced learning theory to handle the third challenge as discussed in the Introduction section, our SETKMC model will reduce to the following ETKMC model:

$$\left\{ \begin{array}{l} \min_{\mathcal{Z}, P} \|\mathcal{Z}\|_{ELTN} + \lambda \sum_{v=1}^V g^v(P^v) \\ s.t. \quad P^v = I - Z^v, \quad v = 1, 2, \dots, V, \\ \quad \mathcal{Z} = \Phi(Z^1, Z^2, \dots, Z^V). \end{array} \right\} \quad (7)$$

This paper will evaluate the clustering performance of both SETKMC in Eq. (6) and ETKMC in Eq. (7) in the next section.

- When we replace our ELTN with the traditional tensor nuclear norm in Eq. (3) and without considering the self-pace learning, our SETKMC will reduce to the method in [28] which is emerged as our main competitor.
- Operator  $\Phi(\cdot)$  aims to construct the representation tensor  $\mathcal{Z}$  by storing all  $Z^v$ , such that the spatial pair-wise correlations and view relationships among multiple views are well encoded. Then tensor  $\mathcal{Z}$  was imposed by the enhanced low-rank tensor norm to explore the low-rank tensor representation. Function  $g^v(\cdot)$  is determined by the kernel type to handle the nonlinearity of multi-view features. Function  $f(\omega, \beta)$  is called the regularizer determining the examples and views to be selected during training. In summary, the proposed SETKMC model addresses the above three challenges simultaneously for multi-view subspace clustering.
- Note that our SETKMC and ETKMC obtain the clustering results by performing the spectral clustering algorithm [16] on the affinity matrix  $S$  defined as  $S = \frac{1}{V} \sum_v (|Z^v| + |Z^{v^T}|)$ .

### B. Optimization of SETKMC

Both SETKMC and ETKMC models can be solved by the traditional ADMM. We first solve the SETKMC model. To address the inseparability of variable  $\mathcal{Z}$ , we introduce one auxiliary variable  $\mathcal{Y}$ . Then, the constrained SETKMC model in Eq. (6) is transformed to the following unconstrained augmented Lagrangian function:

$$\begin{aligned} \mathcal{L}_\rho(\mathcal{Y}, \mathcal{Z}, \mathcal{P}, \omega) = & \|\mathcal{Y}\|_{ELTN} + \lambda \sum_{v=1}^V \sum_{i=1}^n \omega_i^v g^v(P_i^v) + f(\omega, \beta) \\ & + \frac{\rho}{2} \left( \|\mathcal{Z} - \mathcal{Y} + \frac{\Pi}{\rho}\|_F^2 + \sum_v \|I - Z^v - P^v + \frac{\Theta^v}{\rho}\|_F^2 \right), \end{aligned} \quad (8)$$

where  $\Theta$  and  $\Pi$  are Lagrange multipliers.  $\rho$  is the non-negative penalty parameter. Following the alternative update strategy, Eq. (8) can be divided into the following four subproblems:<sup>1</sup>

**Update  $\mathcal{Y}$ :** We minimize Eq. (8) with respect to  $\mathcal{Y}$  and fix the other variables:

$$\mathcal{Y}^* = \operatorname{argmin}_{\mathcal{Y}} \|\mathcal{Y}\|_{ELTN} + \frac{\rho}{2} \|\mathcal{Y} - \mathcal{T}\|_F^2, \quad (9)$$

where  $\mathcal{T} = \mathcal{Z} + \frac{\Pi}{\rho}$ . Since our ELTN in Eq. (4) is nonconvex, we cannot directly yield the closed-form solution of Eq. (9). Inspired by [17], [18], we first rotate  $\mathcal{Y} \in \mathbb{R}^{n \times n \times V}$  into  $\hat{\mathcal{Y}} \in \mathbb{R}^{n \times V \times n}$ , transform Eq. (9) into the frequency domain, and separate  $n$  problems whose  $j$ -th problem is

$$\mathcal{Y}^j = \operatorname{argmin}_{\hat{\mathcal{Y}}^j} \frac{1}{\rho} \sum_{i=1}^V \phi(\sigma_i(\hat{\mathcal{Y}}^j), \gamma) + \frac{1}{2} \|\hat{\mathcal{Y}}^j - \hat{\mathcal{T}}^j\|_F^2, \quad (10)$$

where  $\hat{\mathcal{Y}} = \text{fft}(\bar{\mathcal{Y}}, [], 3)$ .  $\hat{\mathcal{Y}}^j$  is the  $j$ -th frontal slice of  $\hat{\mathcal{Y}}$ . Considering the nonascending order of singular values and the anti-monotone property of gradients of the exponential function, we

have

$$0 \leq \nabla \phi(\sigma_1^k, \gamma) \leq \nabla \phi(\sigma_2^k, \gamma) \leq \dots \leq \nabla \phi(\sigma_V^k, \gamma), \quad (11)$$

$$\phi(\sigma_i(\hat{\mathcal{Y}}^j), \gamma) \leq \phi(\sigma_i^k, \gamma) + \nabla \phi(\sigma_i^k, \gamma)(\sigma_i(\hat{\mathcal{Y}}^j) - \sigma_i^k), \quad (12)$$

where  $\sigma_i^k$  denotes the  $i$ -th singular value of  $\hat{\mathcal{Y}}^j$ .  $\nabla \phi(\sigma_i^k, \gamma)$  is the gradient of  $\phi(\sigma_i(\hat{\mathcal{Y}}^j), \gamma)$  at  $\sigma_i^k$ . Eq. (12) is derived by the super-gradient definition of the concave function [55]. Accordingly, Eq. (10) is relaxed into

$$\begin{aligned} \mathcal{Y}_{k+1}^j = & \operatorname{argmin}_{\hat{\mathcal{Y}}^j} \frac{1}{\rho} \sum_{i=1}^V \phi(\sigma_i^k, \gamma) + \\ & \phi(\sigma_i^k, \gamma)(\sigma_i(\hat{\mathcal{Y}}^j) - \sigma_i^k) + \frac{1}{2} \|\hat{\mathcal{Y}}^j - \hat{\mathcal{T}}^j\|_F^2, \end{aligned} \quad (13)$$

The optimal solution of Eq. (13) can be yielded by the generalized weighted singular value thresholding (WSVT) [55].

**Update  $\mathcal{Z}$ :** We minimize Eq. (8) with respect to  $\mathcal{Z}$  and fix the other variables:

$$\mathcal{Z}^* = \operatorname{argmin}_{\mathcal{Z}} \|\mathcal{Z} - \mathcal{Y} + \frac{\Pi}{\rho}\|_F^2 + \sum_v \|Z^v - M^v\|_F^2, \quad (14)$$

where  $M^v = I - P^v + \frac{\Theta^v}{\rho}$ . Obviously, it is a quadratic optimization problem about  $\hat{Z}^v$ . Setting its derivative to zero, the closed-form solution of Eq. (14) is  $Z^v = 0.5 * (Y^v - \frac{\Pi^v}{\rho} + M^v)$ .

**Update  $P^v$ :** We minimize Eq. (8) with respect to  $P^v$  and fix the other variables:

$$P^{v*} = \operatorname{argmin}_{P^v} \lambda \sum_{i=1}^n \omega_i^v g^v(P_i^v) + \frac{\rho}{2} \|P^v - F^v\|_F^2. \quad (15)$$

where  $F^v = I - Z^v + \frac{\Theta^v}{\rho}$ . The optimal solution can be obtained from [9], [28].

**Update  $\omega$ :** We update the sample weights in the  $v$ -th view  $\omega^v$  by minimizing Eq. (13), which can be separated into  $n$  smaller scale subproblems:

$$\omega_i^{v*} = \operatorname{argmin}_{\omega_i^v} \omega_i^v g^v(P_i^v) + f(\omega_i^v, \beta), \quad (16)$$

where  $f(\omega_i^v, \beta)$  is the popular soft weight regularizer [44] instead of the original hard weights.  $f(\omega_i^v, \beta)$  is defined as

$$f(\omega_i^v, \beta) = \ln(1 + e^{-\frac{1}{\beta}} - \omega_i^v)^{(1+e^{-\frac{1}{\beta}} - \omega_i^v)} + \ln(\omega_i^v) \omega_i^v - \frac{\omega_i^v}{\beta}.$$

The optimal  $\omega_i^{v*}$  of the  $i$ -th sample in the  $v$ -th view can be obtained by setting the gradient with respect to  $\omega_i^v$  to zero, i.e.,

$$\omega_i^{v*} = \frac{1 + e^{-\frac{1}{\beta}}}{1 + e^{g^v(P_i^v) - \frac{1}{\beta}}}. \quad (17)$$

Finally, Lagrangian multipliers  $\Theta$ ,  $\Pi$  and penalty parameter  $\rho$  are updated by

$$\begin{aligned} \Theta^{v*} &= \Theta^v + \rho(I - Z^v - P^v); \\ \Pi^* &= \Pi + \rho(\mathcal{Z} - \mathcal{Y}); \\ \rho^* &= \min\{\tau * \rho, \rho_{max}\}. \end{aligned} \quad (18)$$

<sup>1</sup>For simplicity, the iteration number  $k$  is omitted in the updates of all variables.

**Algorithm 1:** SETKMC for Multi-View Subspace Clustering.

**Input:** multi-view kernel matrices  $\{\mathbf{K}^v\}_{v=1}^V$ ; parameters:  $\gamma, \lambda, \beta$ ;  
**Initialize:**  $\mathcal{Y}, \mathcal{Z}, P, \Theta, \Pi$  initialized to  $\mathbf{0}$ ;  $\rho = 10^{-3}$ ,  $\tau = 1.5, \epsilon = 10^{-7}$ ;  
1: **while** not converged **do**  
2:   **for**  $v = 1$  to  $V$  **do**  
3:     Update  $Y^v, Z^v, P^v, \omega^v$  by Eqs. (13), (14), (15), and (17), respectively;  
4:   **end for**  
5:   Update  $\Theta^v, \Pi$ , and  $\rho$  by Eq. (18);  
6:   Check the convergence condition  
7:

$$\max \left\{ \|\mathcal{Z}_{k+1} - \mathcal{Y}_{k+1}\|_\infty, \|\mathbf{I} - \mathbf{Z}_{k+1}^v - \mathbf{P}_{k+1}^v\|_\infty \right\} \leq \epsilon, \quad (21)$$

8: **end while**

**Output:** Representation tensor  $\mathcal{Z}$ .

where  $1 < \tau < \frac{\sqrt{5}+1}{2}$  is to facilitate the convergence speed [56].  $\rho_{max}$  is the max value of the penalty parameter  $\rho$ . Algorithm 1 summarizes the whole procedures of our SETKMC model in Eq. (6).

### C. Optimization of ETKMC

By introducing one auxiliary variable  $\mathcal{Y}$  to make variable  $\mathcal{Z}$  separable, the augmented Lagrangian function of the ETKMC model in Eq. (7) is

$$\mathcal{L}_\rho(\mathcal{Y}, \mathcal{Z}, P) = \|\mathcal{Y}\|_{ELTN} + \frac{\rho}{2} \|\mathcal{Z} - \mathcal{Y} + \frac{\Pi}{\rho}\|_F^2 + \sum_v \left( \lambda g^v(P^v) + \frac{\rho}{2} \|\mathbf{I} - \mathbf{Z}^v - \mathbf{P}^v + \frac{\Theta^v}{\rho}\|_F^2 \right). \quad (19)$$

Due to the page limitation, we give only the sub-problems of ETKMC different from these of SETKMC.

**Update  $P^v$ :** We minimize Eq. (19) with respect to  $P^v$  and fix the other variables:

$$P^{v*} = \operatorname{argmin}_{P^v} \lambda g^v(P^v) + \frac{\rho}{2} \|P^v - F^v\|_F^2. \quad (20)$$

The difference between Eqs. (15) and (20) is that the ETKMC model does not take weights on  $g^v(P^v)$  into consideration. Meanwhile, solving Eq. (19) does not involve solving the weight subproblem like Eq. (16).

### D. Differences With Existing Works

It can be seen clearly that our approach in Eq. (6) integrates the low-rank tensor representation, the kernel trick, and the self-paced learning into a unified model. From the discussions in Section II and the paradigm, these highly related works include multi-view tensor clustering methods [1], [17]–[19], [32], [33], [45], multi-view kernel clustering ones [9], [28] and self-paced

learning-based ones [43], [44]. One significant difference between our approach and them is that existing methods merely consider to solve only one or two of the above-mentioned three challenges while our SETKMC addresses all three challenges in one unified model. Specifically,

- **Differences with the methods in [1], [9], [17]–[19], [28], [33]:** Although these existing methods utilized the low-rank tensor representation to capture the high-order correlations among all views, the tensor nuclear norm may result in over-penalizing singular values and the biased estimation. Instead of treating all singular values equally, our SETKMC and ETKMC use the enhanced low-rank tensor norm to explicitly consider the salient differences between singular values.
- **Differences with the methods in [32], [45]:** The methods in [32], [45] and our SETKMC and ETKMC share the similar idea using different and enhanced tensor nuclear norms to overcome the biased estimation of the tensor nuclear norm. However, [45] aims to handle the image recovery and background modeling tasks. The work [32] assigns a reasonable weight to the indicator matrix of each view while our SETKMC and ETKMC impose the low-rank constraint on the self-representation tensor. Besides, our SETKMC and ETKMC also integrate the self-paced learning and the kernel trick for multi-view clustering.
- **Differences with the methods in [49]–[52]:** These existing methods and our SETKMC and ETKMC follow the same mechanism to make full use of the consistency and complementary information among multiple views. However, our SETKMC and ETKMC explore the problem of multi-view subspace clustering while these existing they are to solve the problems of large-scale multimedia search, large-scale effective ensemble adversarial attacks, and 3-D object retrieval and classification.

## V. EXPERIMENTAL RESULTS

The main aim of multi-view clustering is to partition unlabeled data points into their corresponding clusters by exploiting the multi-view information. This means that multi-view features and evaluation metrics are indispensable to evaluate the multi-view clustering methods. As a result, we first introduce seven real-world datasets in Section V-A-(1), each of which extracts several different types of features. In Section V-A-(2), we also selected 15 state-of-the-art multi-view clustering methods to verify the effectiveness of the proposed SETKMC and ETKMC algorithms. Following [4], [9], [17], Section V-A-(3) reports six evaluation metrics including accuracy (ACC), normalized mutual information (NMI), adjusted rank index (AR), Fscore, Precision, and Recall. Finally, we analyse the proposed SETKMC including parameter selection and numerical convergence.

### A. Experimental Settings

**(1) Databases:** We selected seven real-world databases as testing data. They are StillDB, ORL, Flowers, COIL-20, Extended YaleB, CMU-PIE-15 and 100leaves. **StillDB** [59] is a



TABLE II  
SUMMARY OF THESE SEVEN REAL MULTI-VIEW DATASETS

Category	Dataset	Instance	View	Cluster
Action image	StillDB	467	3	6
Face image	ORL	400	3	40
Face image	YaleB	640	3	10
Face image	CMU-PIE-15	1020	3	68
Flower image	Flowers	1360	3	17
Object image	COIL_20	1440	3	20
Leaves	100leaves	1600	3	100

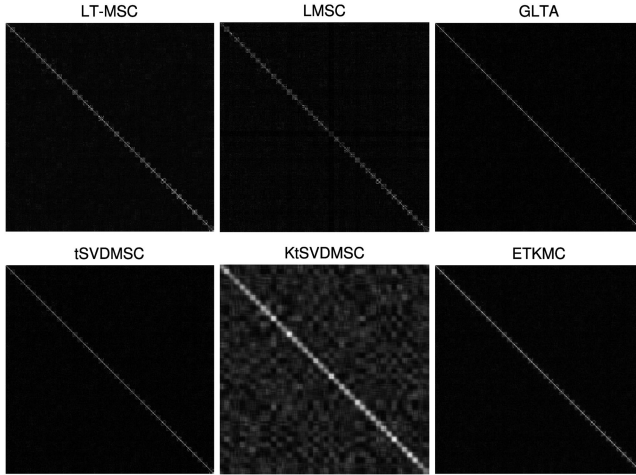


Fig. 1. Comparison of affinity matrices learned by LT-MSC [1], LMSC [7], GLTA [2], tSVD [17], KtSVD [28], and our ETKMC on ORL database. The figure is viewed better in zoomed PDF.

still image database for action recognition including 467 images belonging to 6 categories. **ORL**<sup>2</sup> is a face image database containing 400 face images with 40 different subjects. **Flowers**<sup>3</sup> is a flower database which consists of 1360 samples totally with 17 species. **COIL-20**<sup>4</sup> is an object image database which contains 1440 object images taken by a camera from 72 different angles. **Extended YaleB**<sup>5</sup> is a face image database with 640 face images of 10 people. All of them are captured under different lighting conditions. Following [1], there are three types of features, including 2500  $d$  Intensity, 3304  $d$  LBP, and 6750  $d$  Gabor. **CMU-PIE-15** is a subset of the multi-view face CMU-PIE dataset. It is composed of 68 subjects and 1020 images in total. Following [60], Intensity, LBP and HOG are used as three types of features. **100leaves**<sup>6</sup> is made up of 1600 samples from 100 plant species. For each sample, three types of features are extracted as multi-view data, including the shape descriptor, fine scale margin and texture histogram. We also summarize the details of these seven databases in Table II.

(2) **Baselines:** We compare the proposed SETKMC with the following fifteen state-of-the-art multi-view clustering baselines. The details of them are given as follows: (1) **RMSC** [11]:

<sup>2</sup>[Online]. Available: <http://www.uk.research.att.com/facedatabase.html>

<sup>3</sup>[Online]. Available: <http://www.robots.ox.ac.uk/vgg/data/flowers/>

<sup>4</sup>[Online]. Available: <http://www.cs.columbia.edu/CAVE/software/softlib/>

<sup>5</sup>[Online]. Available: <http://vision.ucsd.edu/~leekc/ExtYaleDatabase/ExtYaleB.html>

<sup>6</sup>[Online]. Available: <https://archive.ics.uci.edu/ml/datasets/One-hundred+plant+species+leaves+data+set>

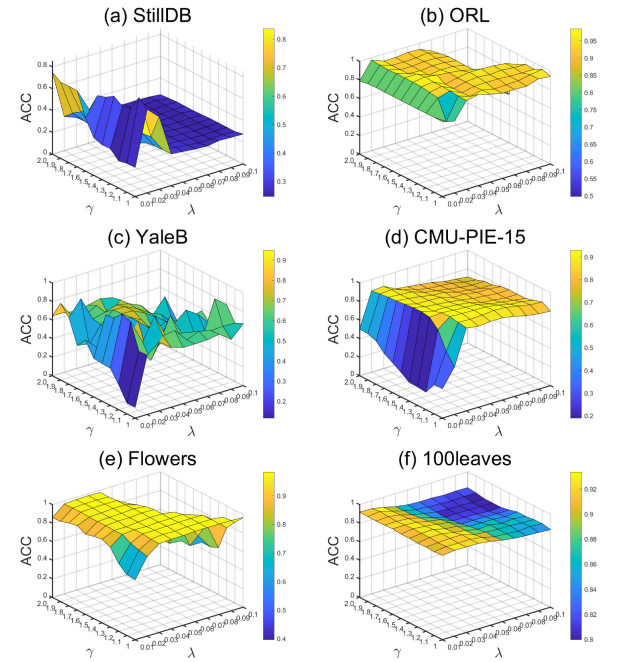


Fig. 2. ACC value of our SETKMC with different combinations of  $\lambda$  and  $\gamma$  on (a) StillDB, (b) ORL, (c) YaleB, (d) CMU-PIE-15, (e) Flowers, and (f) 100leaves database.

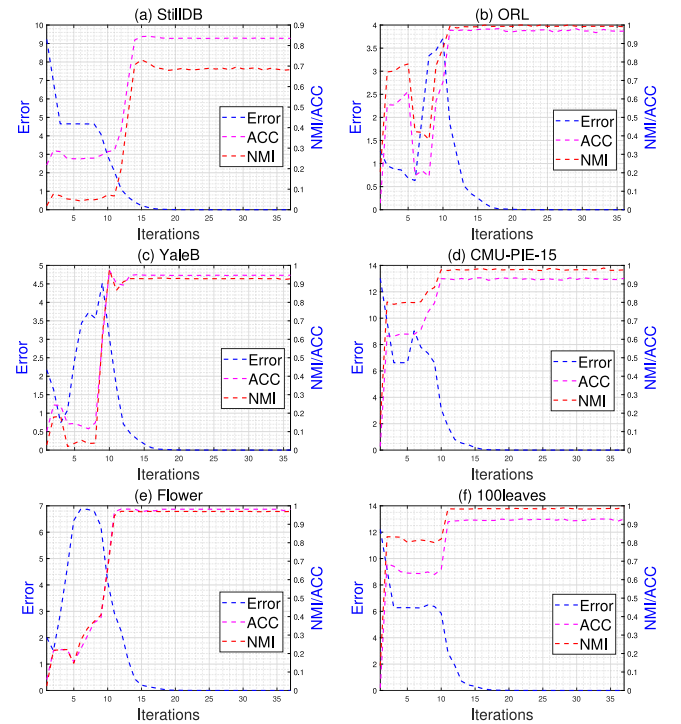


Fig. 3. Convergence (error, ACC and NMI) versus iteration of the proposed SETKMC algorithm.

performing the Markov chain spectral clustering on the shared low-rank representation matrix via low-rank and sparse decomposition; (2) **DiMSC** [6]: adopting the Hilbert-Schmidt Independence criterion to enforce the diversity of different

TABLE III  
MEAN CLUSTERING RESULTS ON STILLDB AND ORL DATASETS

Method	StillDB					
	ACC $\uparrow$	NMI $\uparrow$	AR $\uparrow$	F-score $\uparrow$	Precision $\uparrow$	Recall $\uparrow$
RMSC[11]	0.305 $\pm$ 0.010	0.089 $\pm$ 0.009	0.073 $\pm$ 0.011	0.221 $\pm$ 0.002	0.231 $\pm$ 0.004	0.219 $\pm$ 0.002
DiMSC[6]	0.323 $\pm$ 0.002	0.122 $\pm$ 0.008	0.083 $\pm$ 0.001	0.249 $\pm$ 0.000	0.235 $\pm$ 0.004	0.256 $\pm$ 0.002
LT-MSC[1]	0.342 $\pm$ 0.002	0.136 $\pm$ 0.002	0.090 $\pm$ 0.001	0.252 $\pm$ 0.002	0.243 $\pm$ 0.001	0.261 $\pm$ 0.003
MLAN[57]	0.349 $\pm$ 0.000	0.138 $\pm$ 0.000	0.098 $\pm$ 0.000	0.272 $\pm$ 0.000	0.242 $\pm$ 0.000	0.310 $\pm$ 0.000
LMSC[7]	0.327 $\pm$ 0.003	0.136 $\pm$ 0.003	0.084 $\pm$ 0.011	0.269 $\pm$ 0.005	0.235 $\pm$ 0.007	0.247 $\pm$ 0.012
GLTA[2]	0.366 $\pm$ 0.007	0.126 $\pm$ 0.005	0.102 $\pm$ 0.005	0.262 $\pm$ 0.003	0.251 $\pm$ 0.004	0.275 $\pm$ 0.003
tSVD[17]	0.347 $\pm$ 0.010	0.130 $\pm$ 0.004	0.088 $\pm$ 0.003	0.255 $\pm$ 0.004	0.239 $\pm$ 0.002	0.273 $\pm$ 0.006
ETLMSC[18]	0.604 $\pm$ 0.043	0.520 $\pm$ 0.015	0.423 $\pm$ 0.029	0.523 $\pm$ 0.024	0.518 $\pm$ 0.022	0.528 $\pm$ 0.027
RMSL[58]	0.356 $\pm$ 0.003	0.131 $\pm$ 0.001	0.090 $\pm$ 0.002	0.243 $\pm$ 0.001	0.247 $\pm$ 0.001	0.240 $\pm$ 0.002
KtSVD[28]	0.764 $\pm$ 0.001	0.589 $\pm$ 0.002	0.546 $\pm$ 0.003	0.623 $\pm$ 0.002	0.633 $\pm$ 0.002	0.613 $\pm$ 0.002
JLMVC[9]	0.776 $\pm$ 0.009	0.593 $\pm$ 0.010	0.562 $\pm$ 0.013	0.636 $\pm$ 0.011	0.646 $\pm$ 0.011	0.625 $\pm$ 0.012
LRTG[30]	0.369 $\pm$ 0.002	0.127 $\pm$ 0.001	0.099 $\pm$ 0.001	0.281 $\pm$ 0.001	0.239 $\pm$ 0.001	0.340 $\pm$ 0.001
GNLTA[34]	0.630 $\pm$ 0.030	0.528 $\pm$ 0.019	0.444 $\pm$ 0.022	0.541 $\pm$ 0.018	0.530 $\pm$ 0.018	0.552 $\pm$ 0.019
UGLTL[31]	0.299 $\pm$ 0.002	0.079 $\pm$ 0.006	0.043 $\pm$ 0.001	0.208 $\pm$ 0.001	0.207 $\pm$ 0.001	0.209 $\pm$ 0.001
LTCPCSC[32]	0.371 $\pm$ 0.006	0.160 $\pm$ 0.003	0.105 $\pm$ 0.004	0.256 $\pm$ 0.003	0.260 $\pm$ 0.004	0.253 $\pm$ 0.003
ETKMC	0.788 $\pm$ 0.002	0.618 $\pm$ 0.002	0.582 $\pm$ 0.004	0.653 $\pm$ 0.003	0.663 $\pm$ 0.003	0.643 $\pm$ 0.004
SETKMC	0.805 $\pm$ 0.003	0.654 $\pm$ 0.004	0.618 $\pm$ 0.004	0.683 $\pm$ 0.004	0.693 $\pm$ 0.004	0.673 $\pm$ 0.004
Method	ORL					
	ACC $\uparrow$	NMI $\uparrow$	AR $\uparrow$	F-score $\uparrow$	Precision $\uparrow$	Recall $\uparrow$
RMSC[11]	0.723 $\pm$ 0.007	0.872 $\pm$ 0.012	0.645 $\pm$ 0.003	0.654 $\pm$ 0.007	0.607 $\pm$ 0.009	0.709 $\pm$ 0.004
DiMSC[6]	0.838 $\pm$ 0.001	0.940 $\pm$ 0.003	0.802 $\pm$ 0.000	0.807 $\pm$ 0.003	0.764 $\pm$ 0.012	0.856 $\pm$ 0.004
LT-MSC[1]	0.795 $\pm$ 0.007	0.930 $\pm$ 0.003	0.750 $\pm$ 0.003	0.768 $\pm$ 0.004	0.766 $\pm$ 0.009	0.837 $\pm$ 0.005
MLAN[57]	0.705 $\pm$ 0.022	0.854 $\pm$ 0.018	0.384 $\pm$ 0.010	0.376 $\pm$ 0.015	0.254 $\pm$ 0.021	0.721 $\pm$ 0.020
LMSC[7]	0.877 $\pm$ 0.024	0.950 $\pm$ 0.006	0.839 $\pm$ 0.021	0.843 $\pm$ 0.021	0.805 $\pm$ 0.026	0.884 $\pm$ 0.016
GLTA[2]	0.847 $\pm$ 0.025	0.928 $\pm$ 0.008	0.794 $\pm$ 0.022	0.799 $\pm$ 0.021	0.768 $\pm$ 0.027	0.833 $\pm$ 0.017
tSVD[17]	0.970 $\pm$ 0.003	0.993 $\pm$ 0.002	0.967 $\pm$ 0.002	0.968 $\pm$ 0.003	0.946 $\pm$ 0.004	0.991 $\pm$ 0.003
ETLMSC[18]	0.946 $\pm$ 0.018	0.986 $\pm$ 0.005	0.942 $\pm$ 0.020	0.943 $\pm$ 0.019	0.918 $\pm$ 0.026	0.970 $\pm$ 0.014
RMSL[58]	0.895 $\pm$ 0.010	0.960 $\pm$ 0.002	0.868 $\pm$ 0.011	0.872 $\pm$ 0.011	0.842 $\pm$ 0.018	0.903 $\pm$ 0.004
KtSVD[28]	0.971 $\pm$ 0.021	0.994 $\pm$ 0.007	0.972 $\pm$ 0.022	0.972 $\pm$ 0.022	0.956 $\pm$ 0.027	0.991 $\pm$ 0.017
JLMVC[9]	0.983 $\pm$ 0.018	0.996 $\pm$ 0.004	0.983 $\pm$ 0.018	0.984 $\pm$ 0.017	0.973 $\pm$ 0.028	0.994 $\pm$ 0.006
LRTG[30]	0.933 $\pm$ 0.003	0.970 $\pm$ 0.002	0.905 $\pm$ 0.005	0.908 $\pm$ 0.005	0.888 $\pm$ 0.004	0.928 $\pm$ 0.007
GNLTA[34]	0.910 $\pm$ 0.041	0.977 $\pm$ 0.011	0.904 $\pm$ 0.049	0.906 $\pm$ 0.048	0.861 $\pm$ 0.068	0.957 $\pm$ 0.021
UGLTL[31]	0.924 $\pm$ 0.028	0.970 $\pm$ 0.013	0.912 $\pm$ 0.033	0.913 $\pm$ 0.032	0.887 $\pm$ 0.041	0.941 $\pm$ 0.024
LTCPCSC[32]	0.983 $\pm$ 0.015	0.996 $\pm$ 0.003	0.982 $\pm$ 0.016	0.982 $\pm$ 0.016	0.968 $\pm$ 0.027	0.996 $\pm$ 0.004
ETKMC	0.990 $\pm$ 0.016	0.993 $\pm$ 0.005	0.989 $\pm$ 0.016	0.990 $\pm$ 0.016	0.984 $\pm$ 0.025	0.996 $\pm$ 0.006
SETKMC	0.984 $\pm$ 0.017	0.993 $\pm$ 0.006	0.984 $\pm$ 0.017	0.984 $\pm$ 0.016	0.973 $\pm$ 0.028	0.995 $\pm$ 0.005

views for MVSC; (3) **LT-MSC** [1]: the firstly proposed tensor optimization-based MVSC by the sum of the tensor nuclear norm; (4) **MLAN** [57]: multi-view clustering with adaptive neighbours; (5) **LMSC** [7]: MVSC based on the latent representation; (6) **GLTA** [2]: learning the representation tensor and affinity matrix simultaneously for MVSC; (7) **tSVD** [17]: the second-representative tensor optimization-based MVSC via tensor multi-rank minimization; (8) **ETLMSC** [18]: learning an essential tensor for MVSC; (9) **KtSVD** [28]: the kernel-regularized tSVD; (10) **JLMVC** [9]: jointly learning kernel representation tensor and affinity matrix for MVSC; (11) **RMSL** [58]: the reciprocal multi-layer subspace learning for MVSC; (12) **LRTG** [30]: low-rank tensor graph learning for MVSC; (13) **GNLTA** [34]: generalized nonconvex low-rank tensor approximation for MVSC; (14) **UGLTL** [31]: unified garph and low-rank tensor learning for MVSC; (15) **LTCPCSC** [32]: low-rank tenor constrained co-regularized multi-view spectral clustering. In summary, they are roughly grouped as convex matrix optimization-based methods (RMSC, DiMSC, MLAN, LMSC), convex tensor optimization-based ones (LT-MSC, GLTA, tSVD, ETLMSC, KtSVD, JLMVC, UGLTL), kernel ones (KtSVD, JLMVC), and deep multi-view clustering one (RMSL).

**(3) Evaluation metrics:** Following the experimental setting in [1], [17], we select six popular clustering metrics to evaluate the clustering performance, including accuracy (ACC), normalized mutual information (NMI), adjusted rank index (AR), Fscore, Precision, and Recall. The details of these six metrics can be founded in [17]. Note that the higher values of these six metrics demonstrate the better clustering performance. We followed the parameter settings of all competitors and performed each experiment ten times to eliminate the randomness perturbation since most competitors and the proposed ETKMC and SETKMC models perform the K-means algorithm to yield the clustering results.

### B. Clustering Performance Results

The multi-view clustering results of all methods on those seven databases are reported in Tables III–VI, in which each entry represents the mean values with standard deviations for ten times experiments. For each database, the best results are marked in red while the second-best ones are marked in blue.

It can be clearly observed that the proposed SETKMC and ETKMC achieve the better performance in most cases compared



TABLE IV  
MEAN CLUSTERING RESULTS ON FLOWERS AND COIL-20 DATASETS

Method	Flowers					
	ACC $\uparrow$	NMI $\uparrow$	AR $\uparrow$	F-score $\uparrow$	Precision $\uparrow$	Recall $\uparrow$
RMSC[11]	0.385 $\pm$ 0.016	0.396 $\pm$ 0.014	0.231 $\pm$ 0.019	0.249 $\pm$ 0.011	0.234 $\pm$ 0.012	0.256 $\pm$ 0.010
DiMSC[6]	0.434 $\pm$ 0.014	0.442 $\pm$ 0.011	0.266 $\pm$ 0.009	0.310 $\pm$ 0.008	0.302 $\pm$ 0.007	0.318 $\pm$ 0.010
LT-MSC[1]	0.476 $\pm$ 0.012	0.478 $\pm$ 0.008	0.313 $\pm$ 0.009	0.354 $\pm$ 0.008	0.347 $\pm$ 0.009	0.361 $\pm$ 0.008
MLAN[57]	0.501 $\pm$ 0.008	0.532 $\pm$ 0.003	0.331 $\pm$ 0.010	0.373 $\pm$ 0.009	0.345 $\pm$ 0.010	0.404 $\pm$ 0.006
LMSC[7]	0.442 $\pm$ 0.009	0.444 $\pm$ 0.009	0.275 $\pm$ 0.007	0.318 $\pm$ 0.012	0.312 $\pm$ 0.011	0.325 $\pm$ 0.011
GLTA[2]	0.524 $\pm$ 0.018	0.530 $\pm$ 0.011	0.369 $\pm$ 0.014	0.407 $\pm$ 0.013	0.395 $\pm$ 0.013	0.419 $\pm$ 0.013
tSVD[17]	0.836 $\pm$ 0.005	0.852 $\pm$ 0.002	0.766 $\pm$ 0.002	0.780 $\pm$ 0.002	0.772 $\pm$ 0.002	0.789 $\pm$ 0.002
ETLMSC[18]	0.811 $\pm$ 0.066	0.874 $\pm$ 0.025	0.763 $\pm$ 0.057	0.778 $\pm$ 0.054	0.748 $\pm$ 0.064	0.810 $\pm$ 0.041
RMSL[58]	0.511 $\pm$ 0.006	0.490 $\pm$ 0.007	0.332 $\pm$ 0.010	0.372 $\pm$ 0.005	0.361 $\pm$ 0.008	0.384 $\pm$ 0.011
KtSVD[28]	0.963 $\pm$ 0.030	0.950 $\pm$ 0.013	0.933 $\pm$ 0.033	0.937 $\pm$ 0.031	0.934 $\pm$ 0.037	0.939 $\pm$ 0.024
JLMVC[9]	0.965 $\pm$ 0.010	0.950 $\pm$ 0.007	0.930 $\pm$ 0.015	0.934 $\pm$ 0.014	0.932 $\pm$ 0.016	0.937 $\pm$ 0.011
LRTG[30]	0.550 $\pm$ 0.011	0.555 $\pm$ 0.003	0.389 $\pm$ 0.007	0.427 $\pm$ 0.006	0.405 $\pm$ 0.007	0.452 $\pm$ 0.007
GNLTA[34]	0.845 $\pm$ 0.058	0.895 $\pm$ 0.029	0.808 $\pm$ 0.058	0.819 $\pm$ 0.055	0.791 $\pm$ 0.062	0.850 $\pm$ 0.049
UGLTL[31]	0.977 $\pm$ 0.002	0.965 $\pm$ 0.003	0.953 $\pm$ 0.004	0.956 $\pm$ 0.004	0.955 $\pm$ 0.004	0.956 $\pm$ 0.004
LTCPCSC[32]	0.912 $\pm$ 0.003	0.866 $\pm$ 0.003	0.823 $\pm$ 0.005	0.833 $\pm$ 0.005	0.830 $\pm$ 0.005	0.836 $\pm$ 0.004
ETKMC	0.975 $\pm$ 0.002	0.956 $\pm$ 0.001	0.947 $\pm$ 0.004	0.950 $\pm$ 0.004	0.950 $\pm$ 0.004	0.951 $\pm$ 0.004
SETKMC	0.981 $\pm$ 0.002	0.970 $\pm$ 0.002	0.959 $\pm$ 0.003	0.961 $\pm$ 0.003	0.961 $\pm$ 0.003	0.962 $\pm$ 0.003
Method	COIL-20					
	ACC $\uparrow$	NMI $\uparrow$	AR $\uparrow$	F-score $\uparrow$	Precision $\uparrow$	Recall $\uparrow$
RMSC[11]	0.685 $\pm$ 0.045	0.800 $\pm$ 0.017	0.637 $\pm$ 0.044	0.656 $\pm$ 0.042	0.620 $\pm$ 0.057	0.698 $\pm$ 0.026
DiMSC[6]	0.778 $\pm$ 0.022	0.846 $\pm$ 0.002	0.732 $\pm$ 0.005	0.745 $\pm$ 0.005	0.739 $\pm$ 0.007	0.751 $\pm$ 0.003
LT-MSC[1]	0.804 $\pm$ 0.011	0.860 $\pm$ 0.002	0.748 $\pm$ 0.004	0.760 $\pm$ 0.007	0.741 $\pm$ 0.009	0.776 $\pm$ 0.006
MLAN[57]	0.862 $\pm$ 0.011	0.961 $\pm$ 0.004	0.835 $\pm$ 0.006	0.844 $\pm$ 0.013	0.758 $\pm$ 0.008	0.953 $\pm$ 0.007
LMSC[7]	0.749 $\pm$ 0.018	0.866 $\pm$ 0.006	0.699 $\pm$ 0.025	0.715 $\pm$ 0.023	0.655 $\pm$ 0.041	0.790 $\pm$ 0.017
GLTA[2]	0.878 $\pm$ 0.008	0.945 $\pm$ 0.001	0.869 $\pm$ 0.007	0.875 $\pm$ 0.007	0.856 $\pm$ 0.013	0.895 $\pm$ 0.001
tSVD[17]	0.830 $\pm$ 0.000	0.884 $\pm$ 0.005	0.786 $\pm$ 0.003	0.800 $\pm$ 0.004	0.785 $\pm$ 0.007	0.808 $\pm$ 0.001
ETLMSC[18]	0.877 $\pm$ 0.065	0.947 $\pm$ 0.024	0.862 $\pm$ 0.057	0.869 $\pm$ 0.054	0.830 $\pm$ 0.065	0.914 $\pm$ 0.045
RMSL[58]	0.822 $\pm$ 0.014	0.941 $\pm$ 0.013	0.811 $\pm$ 0.004	0.812 $\pm$ 0.017	0.905 $\pm$ 0.010	0.889 $\pm$ 0.008
KtSVD[28]	0.940 $\pm$ 0.008	0.967 $\pm$ 0.005	0.928 $\pm$ 0.012	0.932 $\pm$ 0.011	0.930 $\pm$ 0.013	0.934 $\pm$ 0.010
JLMVC[9]	0.945 $\pm$ 0.037	0.970 $\pm$ 0.010	0.937 $\pm$ 0.033	0.940 $\pm$ 0.042	0.940 $\pm$ 0.043	0.941 $\pm$ 0.042
LRTG[30]	0.927 $\pm$ 0.000	0.976 $\pm$ 0.000	0.928 $\pm$ 0.000	0.932 $\pm$ 0.000	0.905 $\pm$ 0.000	0.961 $\pm$ 0.000
GNLTA[34]	0.908 $\pm$ 0.032	0.972 $\pm$ 0.019	0.914 $\pm$ 0.027	0.918 $\pm$ 0.025	0.883 $\pm$ 0.035	0.956 $\pm$ 0.019
UGLTL[31]	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000
LTCPCSC[32]	0.990 $\pm$ 0.021	0.990 $\pm$ 0.008	0.984 $\pm$ 0.021	0.985 $\pm$ 0.020	0.982 $\pm$ 0.030	0.989 $\pm$ 0.009
ETKMC	0.968 $\pm$ 0.039	0.996 $\pm$ 0.000	0.962 $\pm$ 0.043	0.964 $\pm$ 0.041	0.957 $\pm$ 0.048	0.970 $\pm$ 0.034
SETKMC	0.993 $\pm$ 0.033	0.994 $\pm$ 0.003	0.987 $\pm$ 0.006	0.988 $\pm$ 0.006	0.987 $\pm$ 0.006	0.988 $\pm$ 0.006

with their competitors. This directly demonstrates the superiority and effectiveness of the proposed enhanced low-rank tensor representation for MVSC.

(1) **Our SETKMC and ETKMC Versus KtSVD:** Specifically, our ETKMC improves by 2.4%, 1.9%, 1.2%, 2.8%, 5.0%, 3.6%, and 0.8% with respect to ACC metric on all databases over the runner-up (KtSVD), respectively. While, the improvement of our SETKMC over KtSVD is around 4.1%, 1.3%, 1.8%, 5.3%, 6.8%, 3.1%, and 1.1% in term of ACC, respectively. The reason is that our proposed enhanced low-rank tensor norm can capture the better low-rankness of the representation tensor against the convex matrix and tensor nuclear norms. In contrast, KtSVD adopted the off-the-shelf tensor nuclear norm without considering the different contributions of different singular values. This can be further verified in Fig. 1 which gives the comparison of affinity matrices learned by several competitors and the proposed ETKMC. Ideally, the block-diagonal elements in affinity matrices should be nonnegative while other elements should be zero. Following this criterion, we can see that LT-MSC,

LMSC, and KtSVD cannot well characterize the cluster structure since they do not satisfy this block diagonal property. The learned affinity matrix of ETKMC has better block diagonal property against these of LT-MSC, LMSC, GLTA, tSVD, and KtSVD.

(2) **Our SETKMC and ETKMC Versus Deep method:** Our SETKMC and ETKMC have yielded the consistently better performance than RMSL, the recently proposed deep MVSC method. This also demonstrates the advantage of the proposed SETKMC and ETKMC. The possible reason may be that the performance of deep learning-based methods often relies heavily on a large number of training samples.

(3) **Kernel trick Versus No-kernel trick:** KtSVD, JLMVC and our SETKMC and ETKMC are based on the kernel trick to deal with the data from nonlinear subspaces. Other competitors belong to no-kernel trick-based methods. As can be seen, KtSVD, JLMVC, our SETKMC and ETKMC are the best four methods among all competitors on most databases. This is because traditional subspace clustering methods such as LT-MSC, LMSC and

TABLE V  
MEAN CLUSTERING RESULTS ON YALEB AND CMU-PIE-15 DATASETS

Method	YaleB					
	ACC $\uparrow$	NMI $\uparrow$	AR $\uparrow$	F-score $\uparrow$	Precision $\uparrow$	Recall $\uparrow$
RMSC[11]	0.210 $\pm$ 0.013	0.157 $\pm$ 0.019	0.060 $\pm$ 0.014	0.155 $\pm$ 0.012	0.151 $\pm$ 0.012	0.159 $\pm$ 0.013
DiMSC[6]	0.615 $\pm$ 0.003	0.636 $\pm$ 0.002	0.453 $\pm$ 0.005	0.504 $\pm$ 0.006	0.481 $\pm$ 0.004	0.534 $\pm$ 0.004
LT-MSC[1]	0.626 $\pm$ 0.010	0.637 $\pm$ 0.003	0.459 $\pm$ 0.030	0.521 $\pm$ 0.006	0.485 $\pm$ 0.001	0.539 $\pm$ 0.002
MLAN[57]	0.346 $\pm$ 0.011	0.352 $\pm$ 0.015	0.093 $\pm$ 0.009	0.213 $\pm$ 0.023	0.159 $\pm$ 0.018	0.321 $\pm$ 0.013
LMSC[7]	0.598 $\pm$ 0.005	0.568 $\pm$ 0.004	0.354 $\pm$ 0.007	0.423 $\pm$ 0.006	0.390 $\pm$ 0.006	0.463 $\pm$ 0.005
GLTA[2]	0.614 $\pm$ 0.004	0.631 $\pm$ 0.006	0.439 $\pm$ 0.007	0.497 $\pm$ 0.006	0.473 $\pm$ 0.006	0.524 $\pm$ 0.006
tSVD[17]	0.652 $\pm$ 0.000	0.667 $\pm$ 0.004	0.500 $\pm$ 0.003	0.550 $\pm$ 0.002	0.514 $\pm$ 0.004	0.590 $\pm$ 0.004
ETLMSC[18]	0.325 $\pm$ 0.011	0.307 $\pm$ 0.021	0.179 $\pm$ 0.019	0.262 $\pm$ 0.017	0.257 $\pm$ 0.017	0.267 $\pm$ 0.017
RMSL[58]	0.914 $\pm$ 0.005	0.833 $\pm$ 0.002	0.814 $\pm$ 0.001	0.832 $\pm$ 0.001	0.826 $\pm$ 0.002	0.838 $\pm$ 0.001
KtSVD[28]	0.896 $\pm$ 0.016	0.893 $\pm$ 0.015	0.813 $\pm$ 0.027	0.832 $\pm$ 0.024	0.821 $\pm$ 0.024	0.842 $\pm$ 0.024
JLMVC[9]	0.910 $\pm$ 0.022	0.897 $\pm$ 0.010	0.832 $\pm$ 0.019	0.849 $\pm$ 0.017	0.837 $\pm$ 0.019	0.860 $\pm$ 0.015
LRTG[30]	0.954 $\pm$ 0.000	0.905 $\pm$ 0.000	0.899 $\pm$ 0.000	0.909 $\pm$ 0.000	0.908 $\pm$ 0.000	0.911 $\pm$ 0.000
GNLTA[34]	0.218 $\pm$ 0.018	0.151 $\pm$ 0.015	0.070 $\pm$ 0.006	0.163 $\pm$ 0.006	0.161 $\pm$ 0.005	0.163 $\pm$ 0.007
UGLTL[31]	0.338 $\pm$ 0.006	0.344 $\pm$ 0.005	0.152 $\pm$ 0.003	0.242 $\pm$ 0.002	0.224 $\pm$ 0.002	0.264 $\pm$ 0.003
LTCPCSC[32]	0.936 $\pm$ 0.006	0.908 $\pm$ 0.006	0.869 $\pm$ 0.010	0.883 $\pm$ 0.010	0.879 $\pm$ 0.010	0.886 $\pm$ 0.009
ETKMC	0.946 $\pm$ 0.002	0.928 $\pm$ 0.002	0.891 $\pm$ 0.003	0.902 $\pm$ 0.003	0.895 $\pm$ 0.003	0.908 $\pm$ 0.003
SETKMC	0.964 $\pm$ 0.017	0.973 $\pm$ 0.008	0.928 $\pm$ 0.035	0.935 $\pm$ 0.031	0.929 $\pm$ 0.035	0.942 $\pm$ 0.027
CMU-PIE-15						
RMSC[11]	0.282 $\pm$ 0.010	0.568 $\pm$ 0.008	0.137 $\pm$ 0.011	0.150 $\pm$ 0.011	0.143 $\pm$ 0.011	0.158 $\pm$ 0.011
DiMSC[6]	0.673 $\pm$ 0.034	0.822 $\pm$ 0.014	0.543 $\pm$ 0.033	0.549 $\pm$ 0.032	0.515 $\pm$ 0.034	0.589 $\pm$ 0.032
LT-MSC[1]	0.732 $\pm$ 0.015	0.856 $\pm$ 0.009	0.615 $\pm$ 0.020	0.621 $\pm$ 0.020	0.583 $\pm$ 0.023	0.664 $\pm$ 0.018
MLAN[57]	0.379 $\pm$ 0.010	0.671 $\pm$ 0.007	0.252 $\pm$ 0.013	0.263 $\pm$ 0.012	0.251 $\pm$ 0.013	0.277 $\pm$ 0.011
LMSC[7]	0.754 $\pm$ 0.023	0.858 $\pm$ 0.009	0.599 $\pm$ 0.017	0.605 $\pm$ 0.013	0.562 $\pm$ 0.015	0.685 $\pm$ 0.021
GLTA[2]	0.708 $\pm$ 0.020	0.844 $\pm$ 0.010	0.586 $\pm$ 0.028	0.592 $\pm$ 0.027	0.555 $\pm$ 0.028	0.634 $\pm$ 0.028
tSVD[17]	0.883 $\pm$ 0.014	0.941 $\pm$ 0.005	0.817 $\pm$ 0.017	0.820 $\pm$ 0.016	0.780 $\pm$ 0.020	0.863 $\pm$ 0.013
ETLMSC[18]	0.645 $\pm$ 0.025	0.808 $\pm$ 0.013	0.525 $\pm$ 0.027	0.532 $\pm$ 0.026	0.494 $\pm$ 0.030	0.577 $\pm$ 0.024
RMSL[58]	0.754 $\pm$ 0.005	0.858 $\pm$ 0.001	0.646 $\pm$ 0.001	0.651 $\pm$ 0.001	0.620 $\pm$ 0.002	0.684 $\pm$ 0.001
KtSVD[28]	0.897 $\pm$ 0.009	0.948 $\pm$ 0.004	0.840 $\pm$ 0.015	0.843 $\pm$ 0.015	0.809 $\pm$ 0.022	0.880 $\pm$ 0.008
JLMVC[9]	0.916 $\pm$ 0.013	0.967 $\pm$ 0.006	0.894 $\pm$ 0.016	0.895 $\pm$ 0.015	0.866 $\pm$ 0.020	0.926 $\pm$ 0.013
LRTG[30]	0.807 $\pm$ 0.009	0.881 $\pm$ 0.002	0.547 $\pm$ 0.033	0.554 $\pm$ 0.032	0.452 $\pm$ 0.041	0.720 $\pm$ 0.003
GNLTA[34]	0.639 $\pm$ 0.030	0.805 $\pm$ 0.012	0.519 $\pm$ 0.024	0.526 $\pm$ 0.024	0.487 $\pm$ 0.025	0.572 $\pm$ 0.025
UGLTL[31]	0.496 $\pm$ 0.010	0.670 $\pm$ 0.008	0.318 $\pm$ 0.008	0.328 $\pm$ 0.007	0.310 $\pm$ 0.008	0.348 $\pm$ 0.009
LTCPCSC[32]	0.718 $\pm$ 0.016	0.843 $\pm$ 0.006	0.478 $\pm$ 0.045	0.487 $\pm$ 0.043	0.398 $\pm$ 0.054	0.635 $\pm$ 0.012
ETKMC	0.933 $\pm$ 0.012	0.973 $\pm$ 0.004	0.919 $\pm$ 0.012	0.919 $\pm$ 0.012	0.888 $\pm$ 0.019	0.953 $\pm$ 0.006
SETKMC	0.928 $\pm$ 0.012	0.977 $\pm$ 0.003	0.913 $\pm$ 0.012	0.914 $\pm$ 0.012	0.881 $\pm$ 0.019	0.951 $\pm$ 0.006

tSVD are under the assumption that data samples are generated from several linear subspaces.

(4) **Tensor optimization Versus Matrix optimization:** Generally, these tensor optimization-based methods (LT-MSC, GLTA, tSVD, ETLMSC, KtSVD, JLMVC, UGLTL, LTCPCSC, our SETKMC and ETKMC) outperform the matrix optimization-based methods including RMSC, DiMSC, LMSC, owing to the capture of multi-dimensional structure of the representation tensor.

### C. Model Analysis

(1) **Parameter selection:** There are three parameters in SETKMC model in Eq. (6), including  $\gamma$ ,  $\lambda$ , and  $\beta$ . Here, we take SETKMC as an example. In all experiments, we set  $\beta = 0.5$ . Thus, two free parameters  $\gamma$  and  $\lambda$  are numerically selected from the range in  $[1, 2]$  and  $[0.01, 0.1]$ , respectively. We conduct a sensitivity test for different combinations of  $\gamma$  and  $\lambda$  in

Fig. 2. One can see that our SETKMC achieves promising performance under different values of parameters  $\gamma$  and  $\lambda$ . For example, ACC values of SETKMC are higher than these of the first four competitors on the ORL and Flowers databases. Generally, SETKMC is relatively robust to parameter  $\gamma$ . The best ACC is obtained when parameter  $\lambda$  falls in a range  $[0.01, 0.05]$ .

(2) **Numerical convergence:** Since it is intractable to guarantee the theoretical convergence when the variables are more than two and the objective function is nonconvex, we investigate the numerical convergence of the proposed SETKMC algorithm. We show the relative errors versus iterations in Fig. 3. Since the ACC and NMI values versus iterations may demonstrate the numerical convergence of the algorithm to some extent, we also show them in Fig. 3. One can see that, with the iterations increasing, the ACC and NMI curves generally go up and achieve stable after reasonable fluctuations. The proposed SETKMC quickly converges within approximately 40 iterations on all testing datasets. This demonstrates that the proposed SETKMC algorithm has good numerical convergence.

TABLE VI  
MEAN CLUSTERING RESULTS ON 100LEAVES DATASET

Method	100leaves					
	ACC↑	NMI↑	AR↑	F-score↑	Precision↑	Recall↑
RMSC[11]	0.711±0.026	0.875±0.008	0.630±0.025	0.634±0.025	0.595±0.027	0.679±0.022
DiMSC[6]	0.721±0.024	0.873±0.007	0.638±0.021	0.641±0.021	0.607±0.025	0.680±0.018
LT-MSC[1]	0.736±0.007	0.870±0.006	0.641±0.012	0.644±0.012	0.615±0.012	0.678±0.013
MLAN[57]	0.883±0.001	0.950±0.001	0.830±0.001	0.823±0.012	0.791±0.018	0.858±0.007
LMSC[7]	0.766±0.015	0.892±0.004	0.686±0.011	0.689±0.011	0.655±0.013	0.725±0.010
GLTA[2]	0.826±0.012	0.926±0.004	0.772±0.011	0.775±0.011	0.740±0.014	0.813±0.010
tSVD[17]	0.923±0.014	0.983±0.002	0.921±0.012	0.922±0.012	0.883±0.018	0.964±0.005
ETLMSC[18]	0.836±0.022	0.962±0.005	0.838±0.020	0.839±0.020	0.782±0.028	0.905±0.014
RMSL[58]	0.730±0.004	0.854±0.004	0.618±0.003	0.621±0.004	0.593±0.002	0.653±0.001
KtSVD[28]	0.919±0.010	0.984±0.002	0.920±0.009	0.921±0.009	0.880±0.013	0.966±0.005
JLMVC[9]	0.908±0.013	0.979±0.004	0.906±0.015	0.907±0.015	0.869±0.017	0.949±0.013
LRTG[30]	0.876±0.010	0.949±0.003	0.831±0.009	0.833±0.009	0.796±0.011	0.873±0.008
GNLTA[34]	0.853±0.020	0.962±0.006	0.846±0.020	0.848±0.020	0.797±0.022	0.906±0.018
UGLTL[31]	0.900±0.009	0.965±0.010	0.885±0.012	0.886±0.013	0.853±0.013	0.922±0.013
LTCPSC[32]	0.947±0.005	0.990±0.001	0.944±0.007	0.945±0.007	0.911±0.013	0.981±0.002
ETKMC	0.927±0.010	0.985±0.003	0.927±0.010	0.928±0.010	0.889±0.014	0.970±0.007
SETKMC	0.930±0.012	0.987±0.005	0.932±0.010	0.932±0.010	0.895±0.014	0.973±0.006

## VI. CONCLUSIONS

In this paper, we developed a novel multi-view subspace clustering (SETKMC) method integrating the enhanced low-rank tensor representation, the kernel trick and the self-paced learning. SETKMC adopted the kernel trick to solve the nonlinearity problem, proposed the enhanced low-rank tensor norm to better approximate the tensor rank, such that the learnt representation tensor can well capture the similarity between data points, and utilized the self-paced learning to gradually involve instances from easy to difficult. An effective algorithm was derived by the alternating direction method of multipliers. Experimental results have demonstrated that our ETKMC and SETKMC achieved performance improvement compared to fifteen state-of-the-art multi-view clustering methods.

## REFERENCES

- [1] 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.
- [2] Y. Chen, X. Xiao, and Y. Zhou, "Multi-view clustering via simultaneously learning graph regularized low-rank tensor representation and affinity matrix," in *Proc. IEEE Int. Conf. Multimedia Expo.* IEEE, 2019, pp. 1348–1353.
- [3] F. Nie, J. Li, and X. Li, "Self-weighted multiview clustering with multiple graphs," in *Proc. Int. Joint Conf. Artif. Intell.*, 2017, pp. 2564–2570.
- [4] Z. Li *et al.*, "Consensus graph learning for multi-view clustering," *IEEE Trans. Multimedia*, May 2021. DOI: 10.1109/TMM.2021.3081930.
- [5] C. Tang *et al.*, "Learning a joint affinity graph for multiview subspace clustering," *IEEE Trans. Multimedia*, vol. 21, no. 7, pp. 1724–1736, Jul. 2019.
- [6] 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.
- [7] 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.
- [8] G. Chao, S. Sun, and J. Bi, "A survey on multi-view clustering," *IEEE Trans. Artif. Intell.*, vol. 2, no. 2, pp. 146–168, Apr. 2021.
- [9] Y. Chen, X. Xiao, and Y. Zhou, "Jointly learning kernel representation tensor and affinity matrix for multi-view clustering," *IEEE Trans. Multimedia*, vol. 22, no. 8, pp. 1985–1997, Aug. 2020.
- [10] G. Liu *et al.*, "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171–184, Jan. 2013.
- [11] 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, pp. 2149–2155.
- [12] X. Xiao, Y.-J. Gong, Z. Hua, and W.-N. Chen, "On reliable multi-view affinity learning for subspace clustering," *IEEE Trans. Multimedia*, Dec. 2020. DOI: 10.1109/TMM.2020.3045259
- [13] 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, Nov. 2013.
- [14] Y. Liu, B. Du, and L. Zhang, "Self-paced subspace clustering," in *Proc. IEEE Int. Conf. Multimedia Expo.*, 2019, pp. 350–355.
- [15] 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, Feb. 2019.
- [16] 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.
- [17] Y. Xie *et al.*, "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.
- [18] J. Wu, Z. Lin, and H. Zha, "Essential tensor learning for multi-view spectral clustering," *IEEE Trans. Image Process.*, vol. 28, no. 12, pp. 5910–5922, Dec. 2019.
- [19] Y. Chen, X. Xiao, Z. Hua, and Y. Zhou, "Adaptive transition probability matrix learning for multiview spectral clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, Mar. 2021. DOI: 10.1109/TNNLS.2021.3059874.
- [20] Y. Chen *et al.*, "Denoising of hyperspectral images using nonconvex low rank matrix approximation," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 9, pp. 5366–5380, Sep. 2017.
- [21] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Weighted nuclear norm minimization with application to image denoising," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 2862–2869.
- [22] Y. Chen, X. Xiao, and Y. Zhou, "Low-rank quaternion approximation for color image processing," *IEEE Trans. Image Process.*, vol. 29, pp. 1426–1439, Sep. 2019. DOI: 10.1109/TIP.2019.2941319.
- [23] 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, Nov. 2016.
- [24] M. Yin, J. Gao, and Z. Lin, "Laplacian regularized low-rank representation and its applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 3, pp. 504–517, Mar. 2016.



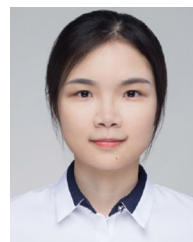
- [25] B. Wang *et al.*, "Adaptive fusion of heterogeneous manifolds for subspace clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 8, pp. 3484–3497, Aug. 2021.
- [26] Y. Xie, W. Zhang, Y. Qu, L. Dai, and D. Tao, "Hyper-laplacian regularized multilinear multiview self-representations for clustering and semisupervised learning," *IEEE Trans. Cybern.*, vol. 50, no. 2, pp. 572–586, Feb. 2020.
- [27] X. Xie, X. Guo, G. Liu, and J. Wang, "Implicit block diagonal low-rank representation," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 477–489, Jan. 2018.
- [28] Y. Xie *et al.*, "Robust kernelized multiview self-representation for subspace clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 2, pp. 868–881, Feb. 2021.
- [29] M. P. Kumar, B. Packer, and D. Koller, "Self-paced learning for latent variable models," in *Proc. Neural Inf. Process. Syst.*, 2010, pp. 1189–1197.
- [30] Y. Chen, X. Xiao, C. Peng, G. Lu, and Y. Zhou, "Low-rank tensor graph learning for multi-view subspace clustering," *IEEE Trans. Circuits Syst. Video Technol.*, Feb. 2021. DOI: 10.1109/TCSVT.2021.3055625.
- [31] J. Wu, X. Xie, L. Nie, Z. Lin, and H. Zha, "Unified graph and low-rank tensor learning for multi-view clustering," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 6388–6395.
- [32] H. Xu, X. Zhang, W. Xia, Q. Gao, and X. Gao, "Low-rank tensor constrained co-regularized multi-view spectral clustering," *Neural Netw.*, vol. 132, pp. 245–252, 2020.
- [33] Y. Jia, H. Liu, J. Hou, S. Kwong, and Q. Zhang, "Multi-view spectral clustering tailored tensor low-rank representation," *IEEE Trans. Circuits Syst. Video Technol.*, Jan. 2021. DOI: 10.1109/TCSVT.2021.3055039.
- [34] Y. Chen, S. Wang, C. Peng, Z. Hua, and Y. Zhou, "Generalized nonconvex low-rank tensor approximation for multi-view subspace clustering," *IEEE Trans. Image Process.*, vol. 30, pp. 4022–4035, Mar. 2021. DOI: 10.1109/TIP.2021.3068646.
- [35] Y. Yang and H. Wang, "Multi-view clustering: A survey," *Big Data Mining Anal.*, vol. 1, no. 2, pp. 83–107, 2018.
- [36] G. Tzortzis and A. Likas, "Kernel-based weighted multi-view clustering," in *Proc. IEEE 12th Int. Conf. Data Mining*, 2012, pp. 675–684.
- [37] S. Yu *et al.*, "Optimized data fusion for kernel k-means clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 5, pp. 1031–1039, May 2012.
- [38] S. Huang, Z. Kang, I. W. Tsang, and Z. Xu, "Auto-weighted multi-view clustering via kernelized graph learning," *Pattern Recognit.*, vol. 88, pp. 174–184, 2019.
- [39] L. Jiang *et al.*, "Self-paced learning with diversity," in *Proc. Neural Inf. Process. Syst.*, 2014, pp. 2078–2086.
- [40] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proc. Int. Conf. Mach. Learn.*, 2009, pp. 41–48.
- [41] Q. Zhao *et al.*, "Self-paced learning for matrix factorization," in *Proc. AAAI Conf. Artif. Intell.*, 2015, pp. 3196–3202.
- [42] D. Zhang, D. Meng, and J. Han, "Co-saliency detection via a self-paced multiple-instance learning framework," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 5, pp. 865–878, May 2017.
- [43] P. Zhou *et al.*, "Self-paced clustering ensemble," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 4, pp. 1497–1511, Apr. 2021.
- [44] C. Xu, D. Tao, and C. Xu, "Multi-view self-paced learning for clustering," in *Proc. Int. Joint. Conf. Artif. Intell.*, 2015, pp. 3974–3980.
- [45] Q. Gao *et al.*, "Enhanced tensor RPCA and its application," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 6, pp. 2133–2140, Jun. 2021.
- [46] C. Zhang *et al.*, "Generalized latent multi-view subspace clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 1, pp. 86–99, Jan. 2020.
- [47] Q. Wang, J. Cheng, Q. Gao, G. Zhao, and L. Jiao, "Deep multi-view subspace clustering with unified and discriminative learning," *IEEE Trans. Multimedia*, Sep. 2020. DOI: 10.1109/TMM.2020.3025666.
- [48] W. Xia, Q. Wang, Q. Gao, X. Zhang, and X. Gao, "Self-supervised graph convolutional network for multi-view clustering," *IEEE Trans. Multimedia*, Jul. 2021. DOI: 10.1109/TMM.2021.3094296.
- [49] X. Lu, L. Zhu, J. Li, H. Zhang, and H. T. Shen, "Efficient supervised discrete multi-view hashing for large-scale multimedia search," *IEEE Trans. Multimedia*, vol. 22, no. 8, pp. 2048–2060, Aug. 2020.
- [50] L. Xie, J. Shen, J. Han, L. Zhu, and L. Shao, "Dynamic multi-view hashing for online image retrieval," in *Proc. Int. Joint. Conf. Artif. Intell.*, 2017, pp. 3133–3139.
- [51] J. Shen and N. Robertson, "BBAS: Towards large scale effective ensemble adversarial attacks against deep neural network learning," *Inf. Sci.*, vol. 569, pp. 469–478, 2021.
- [52] H.-Y. Zhou, A.-A. Liu, W.-Z. Nie, and J. Nie, "Multi-view saliency guided deep neural network for 3-D object retrieval and classification," *IEEE Trans. Multimedia*, vol. 22, no. 6, pp. 1496–1506, Jun. 2020.
- [53] M. Brbić and I. Kopriva, " $l_0$ -motivated low-rank sparse subspace clustering," *IEEE Trans. Cybern.*, vol. 50, no. 4, pp. 1711–1725, Apr. 2020.
- [54] P. Zhou, L. Du, and X. Li, "Self-paced consensus clustering with bipartite graph," in *Proc. Int. Joint. Conf. Artif. Intell.*, 2020, pp. 2133–2139.
- [55] C. Lu, C. Zhu, C. Xu, S. Yan, and Z. Lin, "Generalized singular value thresholding," in *Proc. AAAI Conf. Artif. Intell.*, 2015, pp. 1805–1811.
- [56] S. Boyd *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.
- [57] F. Nie, G. Cai, and X. Li, "Multi-view clustering and semi-supervised classification with adaptive neighbours," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 2408–2414.
- [58] R. Li *et al.*, "Reciprocal multi-layer subspace learning for multi-view clustering," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 8172–8180.
- [59] N. Ikizler, R. G. Cinbis, S. Pehlivan, and P. Duygulu, "Recognizing actions from still images," in *Proc. IEEE Int. Conf. Pattern Recognit.*, 2008, pp. 1–4.
- [60] T. Zhou, C. Zhang, X. Peng, H. Bhaskar, and J. Yang, "Dual shared-specific multiview subspace clustering," *IEEE Trans. Cybern.*, vol. 50, no. 8, pp. 3517–3530, Aug. 2020.



**Yongyong Chen** received the B.S. and M.S. degrees from the Shandong University of Science and Technology, Qingdao, China, in 2014 and 2017, respectively, and the Ph.D. degree from the University of Macau, Macau, China, in 2020. He is currently an Assistant Professor with the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China. His research interests include image processing, data mining, and computer vision.



**Shuqin Wang** received the M.S. degree from the Shandong University of Science and Technology, Qingdao, China, in 2019. She is currently working toward the Ph.D. degree with the Institute of Information Science, Beijing Jiaotong University, Beijing, China. Her research focuses on multiview learning.



**Xiaolin Xiao** received the B.E. degree from Wuhan University, Wuhan, China, in 2013 and the Ph.D. degree from the University of Macau, Macau, China, in 2019. She is currently a Postdoc Fellow with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. Her research interests include multiview learning and color image processing and understanding.



**Youfa Liu** received the M.S. degree in mathematics from the Wuhan Institute of Physics and Mathematics, Chinese Academy of Sciences, Beijing, China, in 2017 and the Ph.D. degree in computer science from Wuhan University, Wuhan, China, in 2020. He is currently a Lecture with the College of Informatics, Huazhong Agricultural University, Wuhan, China. He has authored or coauthored some papers in top journals, such as IEEE TIP, IEEE TNNLS, and INS. His research interests include machine learning and computer vision.



**Zhongyun Hua** (Member, IEEE) received the B.S. degree in software engineering from Chongqing University, Chongqing, China, in 2011, and the M.S. and Ph.D. degrees in software engineering from the University of Macau, Macau, China, in 2013 and 2016, respectively. He is currently an Associate Professor with the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, Shenzhen, China. He has authored or coauthored more than 30 technical papers at prestigious international journals and conferences, including IEEE TSP, TCYB, TSMC-S, TCAS-I, TMM, and TIE. His current research interests include chaotic system, multimedia security, and image processing.



**Yicong Zhou** (Senior Member, IEEE) received the B.S. degree in electrical engineering from Hunan University, Changsha, China, and the M.S. and Ph.D. degrees in electrical engineering from Tufts University, Medford, MA, USA. He is currently a Professor with the Department of Computer and Information Science, University of Macau, Macau, China. His research interests include image processing, computer vision, machine learning, and multimedia security. Dr. Zhou is a fellow of the Society of Photo-Optical Instrumentation Engineers (SPIE) and was recognized as one of the World's Top 2% Scientists on the Stanford University Releases List and one of the Highly Cited Researchers in 2020. He was the recipient of the Third Price of Macao Natural Science Award as a sole winner in 2020 and the Co-Recipient in 2014. Since 2015, she has been a leading Co-Chair of the Technical Committee on Cognitive Computing in the IEEE Systems, Man, and Cybernetics Society. He is an Associate Editor for the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, and four other journals.