IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 28, 2022, accepted May 15, 2022, date of publication May 18, 2022, date of current version May 26, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3176436

Robust Kernelized Multiview Clustering Based on High-Order Similarity Learning

YANYING MEI[1], ZHENWEN REN[1,2], (Member, IEEE), BIN WU[1], TAO YANG[1], AND YANHUA SHAO[1]

[1]School of Information Engineering, Southwest University of Science and Technology, Mianyang 621010, China
[2]State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210008, China

Corresponding author: Zhenwen Ren (rzw@njust.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 62106209; in part by the Sichuan Science and Technology Program under Grant 2021YJ0083; in part by the State Key Laboratory Foundation for Novel Software Technology of Nanjing University under Grant KFKT2021B23; and in part by the Open Project Program of the State Key Laboratory of CAD&CG, Zhejiang University, under Grant A2217.

ABSTRACT This paper explores the robust kernelized multi-view clustering (MVC) for nonlinear data. The existing MVC methods aim to excavate the complementary and consensus information from multi-view data lies in the linear space for clustering. However, in real-world scenarios, data points usually lie in multiple nonlinear spaces, leading to undesirable clustering results. To this end, we propose a robust kernelized MVC method based on high-order similarity learning (RKHSL), to jointly learn the local structure affinities in original space, the nonlinear affinities in mapping kernel space, and the high-order correlations in tensor space. Specifically, the first-order similarity (FOS) is learned to excavate the local structure affinities and the second-order similarity (SOS) is learned in the high-dimensional kernel space to excavate the nonlinear affinities of data points. Afterwards, the third-order similarity (TOS) based on low-rank tensor is learned to excavate the global consistency from multiple views. Extensive experiments on five commonly benchmark datasets show that the proposed method outperforms state-of-the-art methods in most scenarios and is capable of revealing a reliable affinity graph structure concealed in different data points.

INDEX TERMS Multi-view clustering, high-order similarity, low-rank tensor learning, kernel method.

I. INTRODUCTION

As for the new coming online data rising so fast daily [1], [2], interest in data clustering is growing rapidly. Clustering as a preprocessing method of unlabeled data has been extensively applied in data preprocessing, statistics, computer science and other fields [3]–[5]. In real-world scenarios, data points sampled from the same object have multiple heterogeneous features, multi-view clustering (MVC) is commonly used to combine the diverse and consistent information in each view for clustering [6]–[8].

For effectively learn the local structural affinities among data points to deal with the noise and corruption interference, various multi-view graph-based clustering (MVGC) approaches have been proposed [6], [9]–[11], most of which first obtain a k-nearest neighbor graph [9], a self-expression affinity graph [7], or an adaptive neighbor graph [12], and then the final clustering is computed by using the obtained affinity graph. This indicates that the learned affinity graph is very important for the clustering result.

Although these MVGC methods have achieved promising performances, they only study the local structure affinities of pairwise data points by the first-order similarity (FOS) learning. However, the adjacent structures of pairwise data points are also important. Motivated by [13], [14], we explore the adjacent structure affinities of pairwise data points with the second-order similarity (SOS) learning. Moreover, these MVGC methods neglect the higher-order correlations among different views, so we hope to learn the three-order similarity (TOS) of multiple views to excavate the consensus information and view-specific information in terms of tensors.

The proposed method is motivated by the tensor-based MVC model, for example, t-SVD-MSC [15], and UGLTL [16]. Even these methods have achieved outstanding clustering performances, they are mainly based on linear data processing in the original space and may fail when dealing with

The associate editor coordinating the review of this manuscript and approving it for publication was Qilian Liang.

nonlinear data. In addition to the linear data, [17] extended the tensor-based MVC to a kernel method by introducing the "kernel trick," which gains significant improvements in handling nonlinear data. However, it merely considers the high-order affinities among data points.

Generally, based on the graph learning, the affinity for each data point is assumed to be the linear representation coefficients on the other data points restricted to low-rank or sparsity constraint. However, this assumption may not always be true in many real-world scenarios, where the data points are normally considered to be obtained from the nonlinear spaces. Commonly, manifold constraint methods [18], [19] and kernel mapping methods [20] are used to deal with nonlinear data. For kernel mapping method, since data points obtained from a nonlinear low-dimensional space are usually concealed in a high-dimensional kernel space [21], [22], therefore, the data points can be transformed from the original nonlinear space into a high-dimensional kernel space by using kernel mapping, and the original features are preserved in the high-dimensional space [20], [23].

Though the methods mentioned above have obtained outstanding performance for clustering, we find that: 1) for the limitations of the models, the tensor-based methods cannot exploit the SOS among data points, leading to unsatisfying clustering performance; 2) MVGC-based methods cannot exploit the TOS among multiple views, the specific information in each view and consensus information among multiple views cannot be efficiently maintained; and 3) these methods can not effectively deal with the nonlinear data. Thus, a new multi-view graph learning paradigm is urgently needed to improve these MVGC methods.

To solve the above three problems, a MVGC method is proposed, namely robust kernelized multi-view clustering based on high-order similarity learning (RKHSL). The overview of RKHSL is shown in Figure 1. Specifically, the FOS is learned to excavate the local structure affinities of data points in the original space, and the SOS is learned in the high-dimensional kernel spaces to excavate the nonlinear affinities of data points. Afterwards, the TOS of multiple views is learned in a three-order tensor by introducing tensor rotation and tensor Singular Value Decomposition (t-SVD). After that, the optimal consensus affinity graph can be achieved by learning the FOS, SOS and TOS simultaneously. Finally, the clustering results are achieved by using spectral clustering on the learned optimal affinity graph.

In summary, the main contributions of our work are given as below:

- A new high-order similarity learning paradigm is proposed. Unlike the common similarity learning, the high-order similarity learning can fully exploit the potential affinities among data points for data clustering.
- A robust kernelized MVC method, RKHSL, is proposed for nonlinear data clustering, which deeply exploits the high-order similarity among data points. Moreover, an optimization solver is made to solve the objective function.
- Compared with the current state-of-the-art methods, the proposed RKHSL performs well on several benchmark datasets for various applications.

The followings of this paper are structured as follows. The related work is introduced in section II. Section III introduces the notations and preliminaries. Section IV proposes our method RKHSL, optimization and complexity analysis. The experiment is discussed in section V. Finally, section VI concludes with some discussions.

II. RELATED WORK

The latest developments on MVGC and some linked techniques to dispose of nonlinear data will be discussed in this section.

Based on the graph learning model, some MVGC methods have been developed [6], [24], [25]. For example, Nie *et al.* [26] proposed to automatically learn the optical weights for each affinity graph to cluster. Wang *et al.* [6] introduced a MVGC method which learned the unified affinity graph and each affinity graph interactively. Chen *et al.* [24] explore to learn a latent embedding space from all views for clustering. Currently, tensor-based MVGC [27]–[33] have attracted widely attention by stacking all the affinity graphs as a 3-order tensor. Especially, the high-order correlations among multiple views can be automatically learned by using the low-rank constraint of the tensors. For instance, Zhang *et al.* [34] first proposed a low-rank tensor for MVGC based on self-expression model. Xie *et al.* [15] further expanded it by using tensor rotation and t-SVD [35]. Zhao *et al.* [36] proposed a multi-view spectral clustering (MVSC) method based on adaptive graph learning and tensor Schatten p-norm constraint. Zhang et al. [37] introduced a low-rank tensor constrained MVC based on subspace representation learning. Chen *et al.* [28] proposed a multi-view subspace clustering (MSC) based on low-rank tensor graph learning. Xu *et al.* [38] proposed a weighted tensor nuclear constrained MVC based on co-regularized learning. Wang *et al.* [30] proposed a error-robust low-rank tensor approximation method for MVC.

Generally, there are two methods adopted for dealing with the data lies in nonlinear spaces. One is the manifold operation [18], which handles nonlinear data in manifold space. Especially, if two data points are close in the original nonlinear space, after some basic projection, the new representations of them are also close. Grassmann manifolds can be considered as a low-dimensional nonlinear manifolds space embedding in a high-dimensional Euclidean space. There are several classic representations on Grassmann manifolds. For example, Wang *et al.* [39] proposed a low-rank constraint model with non-linear learning on Grassmann manifolds. Later, Wang *et al.* [40] adopted a double nuclear norms constraint instead of the single nuclear norm constraint based on low-rank model for clustering on Grassmann manifolds. Guo *et al.* [41] introduced a

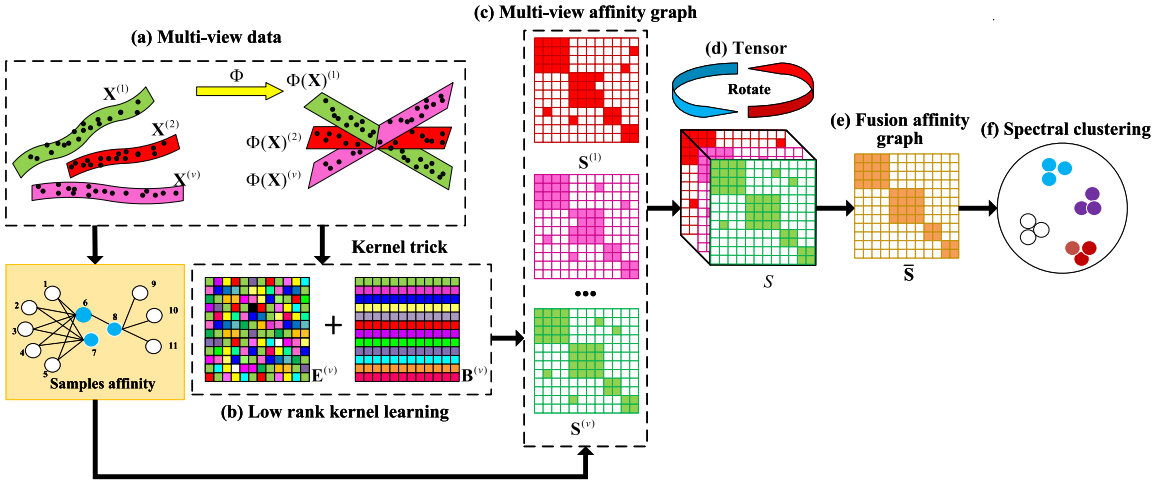


FIGURE 1. Overview of the proposed RKHSL method. (a) Given the multi-view data, (b) the multi-view data is mapped into the higher-dimensional kernel space, (c) V affinity graphs of each corresponding to a view are first constructed by FOS learning in the original space and SOS learning in the kernel space. Then, (d) the resulting affinity graphs are stacked and rotated to form a tensor $\mathcal{S}$, (e) the fusion affinity graph $\bar{\mathbf{S}}$ is obtained by averaging all frontal slices of tensor $\mathcal{S}$. Finally, (f) spectral clustering is used to obtain the final clustering result.

MSC based on low-rank representation learning on product Grassmann manifolds. Wang *et al.* [42] proposed to learn an adaptive neighborhood graph for MSC on Grassmann manifolds.

The other way to handle nonlinear data is the kernel mapping approach, as we can see that the nonlinear mapping by using "kernel trick" can divide the data points in the same allocation, making them linearly separable [43]. For example, Vidal [4] *et al.* applied the "kernel trick" on SSC [44] to deal with the nonlinear data. Alternatively, Zhang *et al.* [21] proposed a kernel MSC by using a low-rank kernel mapping. Later, Zhang *et al.* [45] proposed a kernel MSC by automatically learning the optimal weights for each affinity graph. Hajjar *et al.* [46] proposed a one-step kernel MVC. Qiu *et al.* [47] proposed a ensemble-based clustering with "kernel trick." Zhang *et al.* [48] introduced a robust MVC based on multiple kernel low-rank representation learning. Ren *et al.* [23] proposed a multiple kernel clustering based on local graph and low-rank kernel simultaneously learning. All these methods cannot make full use of the high-order correlations among multiple heterogeneous views.

III. NOTATIONS AND PRELIMINARIES

A. NOTATIONS

In our paper, the 3-order tensors are denoted by bold calligraphy letters (e.g., $\mathcal{P} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$), the matrices are denoted by bold capital letters (e.g., $\mathbf{P}$), the vectors are denoted by bold small letters (e.g., $\mathbf{p}$), and the scalars are denoted by Greek letters (e.g., α, β, λ). We use $\mathbf{p}_j$ and p_{ij} to represent the j-th column vector and the (i, j)-th entry of matrix $\mathbf{P}$, respectively. The related representations for tensor $\mathcal{P} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ are shown in Table 1.

B. PRELIMINARIES

To help us clearly understand the computation of tensors, usually, we first present some related definitions about tensors.

Definition 1 (T-Product): Given tensor $\mathcal{P} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ and tensor $\mathcal{Q} \in \mathbb{R}^{N_2 \times N_4 \times N_3}$, the t-product between them is defined as $\mathcal{P} * \mathcal{Q} \in \mathbb{R}^{N_1 \times N_4 \times N_3}$, i.e.,

$$\mathcal{P} * \mathcal{Q} = \text{fold}(\text{bcirc}(\mathcal{P})\,\text{bvec}(\mathcal{Q})). \tag{1}$$

Definition 2 (f-Diagonal Tensor): If each frontal slice of a tensor is a diagonal matrix, the tensor is considered as f-diagonal.

Definition 3 (Identity Tensor): If the first frontal slice of a tensor $\mathcal{I} \in \mathbb{R}^{N_1 \times N_1 \times N_3}$ satisfies $N_1 \times N_1$ identity matrix and all the other frontal slices satisfy zero matrix, the tensor is considered as identity tensor.

Definition 4 (Orthogonal Tensor): The orthogonal tensor $\mathcal{P} \in \mathbb{R}^{N_1 \times N_1 \times N_3}$ satisfies

$$\mathcal{P}^{\mathrm{T}} * \mathcal{P} = \mathcal{P} * \mathcal{P}^{\mathrm{T}} = \mathcal{I}. \tag{2}$$

Definition 5 (t-SVD): The t-SVD of a tensor $\mathcal{P} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ can be expressed as

$$\mathcal{P} = \mathcal{U} * \mathcal{Q} * \mathcal{V}^{\top}, \tag{3}$$

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

Definition 6 (t-SVD Based Tensor Nuclear Norm): The t-SVD of a tensor $\mathcal{P} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ based on tensor nuclear norm $\|\mathcal{P}\|_{\circledast}$ is given by the sum of singular values of all the frontal slices of $\mathcal{P}_f$:

$$\|\mathcal{P}\|_{\circledast} = \sum_{k=1}^{N_3} \left\|\mathcal{P}_f^{(k)}\right\|_* = \sum_{i=1}^{\min(N_1,N_2)} \sum_{k=1}^{N_3} \left|\mathcal{Q}_f^{(k)}(i,i)\right|, \tag{4}$$

TABLE 1. The representations of the tensor.

Definition	Representation
Mode-1 fiber	$\mathcal{P}(:, j, k)$
Mode-2 fiber	$\mathcal{P}(i, :, k)$
Mode-3 fiber	$\mathcal{P}(i, j, :)$
The k-th frontal slice	$\mathcal{P}(:, :, k)$, $\mathbf{P}_f^k$
The i-th horizontal slice	$\mathcal{P}(i, :, :)$, $\mathbf{P}_h^i$
The j-th lateral slice	$\mathcal{P}(:, j, :)$, $\mathbf{P}_l^j$
The (i, j, k)-th element	$\mathcal{P}(i, j, k)$
$\mathcal{P}_f = \text{fft}(\mathcal{P}, [\,], 3)$	$\mathcal{P}_f$
$\mathcal{P} = \text{ifft}(\mathcal{P}_f, [\,], 3)$	$\mathcal{P}$
The block vectorizing	$\text{bvec}(\mathcal{P}) = \left[\mathbf{P}^{(1)}; \mathbf{P}^{(2)}; \cdots; \mathbf{P}^{(N_3)}\right] \in \mathbb{R}^{N_1 N_3 \times N_2}$
The inverse operation of bvec($\mathcal{P}$)	$\text{fold}(\text{bvec}(\mathcal{P})) = \mathcal{P}$
The block diagonal matrix of $\mathcal{P}$	$\text{bdiag}(\mathcal{P}) \in \mathbb{R}^{N_1 N_3 \times N_2 N_3}$
The corresponding block circulant matrix of $\mathcal{P}$	$\text{bcirc}(\mathcal{P}) \in \mathbb{R}^{N_1 N_3 \times N_2 N_3}$

where $\mathcal{Q}_f^{(k)}$ is computed by the SVD, $\mathcal{P}_f^{(k)} = \mathcal{U}_f^{(k)} \mathcal{Q}_f^{(k)} \mathcal{V}_f^{(k)\top}$ of the frontal slices of $\mathcal{P}_f$.

Definition 7 (Tensor Transpose): The transpose tensor of the tensor $\mathcal{P} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ is defined as $\mathcal{P}^\top \in \mathbb{R}^{N_2 \times N_1 \times N_3}$, which can be obtained by transposing all the frontal slices of $\mathcal{P}$.

Definition 8 (3-order Tensor): Given V matrices $\mathbf{P}^{(v)} \in \mathbb{R}^{N_1 \times N_1}$, the 3-order tensor $\mathcal{P}^* \in \mathbb{R}^{N_1 \times N_1 \times V}$ is constructed by stacking all the V matrices, which is defined as

$$\mathcal{P}^* = \text{bvfold}([\mathbf{P}^{(1)}; \cdots; \mathbf{P}^{(V)}]). \tag{5}$$

Definition 9 (Tensor Rotation): The rotation of the tensor $\mathcal{P}^* \in \mathbb{R}^{N_1 \times N_1 \times N_3}$ is defined as

$$\mathcal{P} = \text{rotate}(\mathcal{P}^*), \tag{6}$$

where $\mathcal{P} \in \mathbb{R}^{N_1 \times N_3 \times N_1}$. Note here that rotate is a shift function.

IV. PROPOSED METHOD

In this section, we propose the RKHSL, its optimization strategy and complexity analysis.

A. FIRST-ORDER SIMILARITY (FOS) LEARNING

The FOS represents the direct similarity s_{ij} of two data points $\mathbf{x}_i$ and $\mathbf{x}_j$, which is the first and foremost measure of similarity between two data points. According to [6], [49], the FOS learning model is defined as

$$\begin{aligned} &\min_{\mathbf{S}} \sum_{i,j=1}^{N} \left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2 s_{ij} \\ &\text{s.t. } \mathbf{S}^\top = \mathbf{S}, \mathbf{0} \le \mathbf{S} \le \mathbf{1}, \end{aligned} \tag{7}$$

where s_{ij} in affinity graph $\mathbf{S} \in \mathbb{R}^{N \times N}$ denotes the affinity between $\mathbf{x}_i$ and $\mathbf{x}_j$, and N denotes the number of data points.

According to [23], we introduce a complete graph $\mathbf{D}$, in which all the data points are treated as nodes, and the affinities among data points are expressed as edge weights.

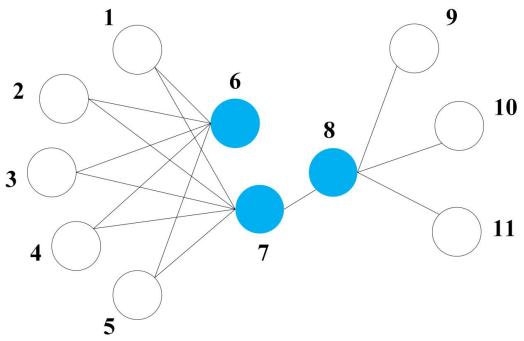


FIGURE 2. A toy example of the FOS and SOS among data points. Data points 7 and 8 should be first-order similar as they are directly connected. Data points 6 and 7 should be second-order similar as they share the similar neighbors.

So we have

$$\begin{aligned} &\min_{\mathbf{S}} \sum_{i,j=1}^{N} \left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2 s_{ij} = \min_{\mathbf{S}} \text{Tr}(\mathbf{D}^\top \mathbf{S}) \\ &\text{s.t. } \mathbf{S}^\top = \mathbf{S}, \mathbf{0} \le S \le \mathbf{1}, \end{aligned} \tag{8}$$

where $\mathbf{D}$ represents a complete graph with $d_{ij} = \left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2$. The smaller the d_{ij} in $\mathbf{D}$ is, the greater the similarity s_{ij} is.

B. SECOND-ORDER SIMILARITY (SOS) LEARNING

The SOS between pairwise data points represents the similarity of their adjacent structures. Given $\boldsymbol{s_i} = [s_{i1}, \cdots, s_{iN}]$, which represents the FOS between $\mathbf{x}_i$ and other data points. Then, the SOS between $\mathbf{x}_i$ and $\mathbf{x}_j$ denotes the similarity between $\boldsymbol{s_i}$ and $\boldsymbol{s_j}$ [49]. Figure 2 shows a toy example of the data points relation based on FOS and SOS in an affinity graph. Note that the data points relation based on FOS represents the pairwise similarity between data points, while the data points relation based on SOS represents the similarity between a data point and its adjacent point set.

The SOS learning model is given by

$$\min_{\mathbf{S}} \sum_{i=1}^{N} \left\| \mathbf{x}_i - \sum_{j=1}^{N} s_{ij}\mathbf{x}_j \right\|_2^2 \quad \text{s.t. } \mathbf{S}^\top = \mathbf{S}, \mathbf{0} \le S \le \mathbf{1}. \tag{9}$$

With (9), we keep the data point $\mathbf{x}_i$ close to its neighbors. As shown in Figure 2, data points 6 and 7 share the same adjacent point set $\{1, 2, \cdots, 5\}$, (9) guarantees that data points 6 is close to data points set $\{1, 2, \cdots, 5\}$, and data point 7 is also close to $\{1, 2, \cdots, 5\}$, so data points 6 and 7 will be close even if they are not directly connected. This implicitly preserves the affinity based on SOS of two unconnected data points.

According to [49], we have

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

where $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N]$ denotes the data matrix consists of N data points and $\operatorname{Tr}(\cdot)$ denotes the trace of the matrix.

C. THIRD-ORDER SIMILARITY (TOS) LEARNING

The TOS is proposed to excavate the third-order correlations among multiple views [16]. Given a tensor $\boldsymbol{\mathcal{S}}$, which is constructed by combing all input affinity graphs $\mathbf{S}^{(v)}$. The nuclear norm $\|\cdot\|_{\circledast}$ is adopted to constrain the 3-order tensor $\boldsymbol{\mathcal{S}}$. The TOS learning model is given as

$$\begin{aligned} \min_{\mathbf{S}^{(v)}} &\sum_{v=1}^{V} \Big(\operatorname{Tr}(\mathbf{X}^{(v)}(\mathbf{I}-\mathbf{S}^{(v)})(\mathbf{I}-\mathbf{S}^{(v)})^\top \mathbf{X}^{(v)\top}) \\ &+ \operatorname{Tr}(\mathbf{D}^{(v)\top}\mathbf{S}^{(v)}) \Big) + \beta \|\boldsymbol{\mathcal{S}}\|_{\circledast} \\ \text{s.t. } &\mathbf{S}^{(v)\top} = \mathbf{S}^{(v)}, \mathbf{0} \le \mathbf{S}^{(v)} \le \mathbf{1}, \\ \boldsymbol{\mathcal{S}} = &\text{ rotate}\left(\text{bvfold}([\mathbf{S}^{(1)}; \cdots; \mathbf{S}^{(V)}])\right), \end{aligned} \tag{11}$$

where $\mathbf{X}^{(v)}$ and $\mathbf{S}^{(v)}$ represent the data matrix and affinity graph in the v-th view, respectively, the tensor $\boldsymbol{\mathcal{S}} \in \mathbb{R}^{N\times V\times N}$ denotes the rotated tensor which is constructed by collecting all affinity graph $\mathbf{S}^{(v)}$, V represents the number of views, and $\beta > 0$ is the trade-off parameter. The tensor rotation is used to effectively capture the common information among multiple heterogeneous views.

As we all know, a good affinity graph with strict block diagonal property is benefit for similarity learning clustering [20], [34], [50], [51]. Therefore, we aim to learn an optimal affinity graph with strict block diagonal property in our method. According to [20], if the clustering indicator matrix satisfies $\mathbf{Q} := \{\mathbf{Q} \mid \mathbf{Q} \in \mathbb{R}^{N\times c}, \mathbf{Q}^\top\mathbf{Q} = \mathbf{I}\}$, $\mathbf{Q}\mathbf{Q}^\top$ is strictly block diagonal. Then, we enforce the affinity graph $\mathbf{S}$ to be strictly block diagonal by introducing Theorem 1.

Theorem 1 [20]: $\mathbb{S}_1 := \{\mathbf{S} \mid \mathbf{S} = \mathbf{Q}\mathbf{Q}^\top, \mathbf{Q}^\top\mathbf{Q} = \mathbf{I}\}$, $\mathbf{Q} \in \mathbb{R}^{N\times c}$, *and* $\mathbb{S}_2 := \left\{\mathbf{S} \mid \mathbf{S} = \mathbf{S}^\top, \operatorname{Tr}(\mathbf{S}) = c, \mathbf{0} \le \mathbf{S} \le \mathbf{1}\right\}$, $\mathbb{S}_2$ *is the convex hull of* $\mathbb{S}_1$, *and* $\mathbb{S}_1$ *is exactly the set of extreme points of* $\mathbb{S}_2$.

Hereto, optimal affinity graph $\mathbf{S}$ can be learned by

$$\begin{aligned} \min_{\mathbf{S}^{(v)}} &\sum_{v=1}^{V} \Big(\operatorname{Tr}(\mathbf{X}^{(v)}(\mathbf{I}-\mathbf{S}^{(v)})(\mathbf{I}-\mathbf{S}^{(v)})^\top \mathbf{X}^{(v)\top}) \\ &+ \operatorname{Tr}(\mathbf{D}^{(v)\top}\mathbf{S}^{(v)}) \Big) + \beta \|\boldsymbol{\mathcal{S}}\|_{\circledast} \\ \text{s.t. } &\operatorname{Tr}(\mathbf{S}^{(v)}) = c, \mathbf{S}^{(v)\top} = \mathbf{S}^{(v)}, \mathbf{0} \le \mathbf{S}^{(v)} \le \mathbf{1}, \\ &\boldsymbol{\mathcal{S}} = \text{ rotate}\left(\text{bvfold}([\mathbf{S}^{(1)}; \cdots; \mathbf{S}^{(V)}])\right), \end{aligned} \tag{12}$$

where c represents the number of classes.

It can be seen that the trace of the affinity graph is decided by the number of clusters, which plays an important role for MVC [29].

D. ROBUST KERNELIZED MULTI-VIEW CLUSTERING WITH HIGH-ORDER SIMILARITY LEARNING

Although (12) has achieved better performance, it cannot effectively deal with nonlinear data. To solve this problem, some kernel-based approaches are proposed in which the nonlinear data is mapped into a high-dimensional kernel space and the linear operations are performed in this kernel space. The "kernel trick" is commonly used in kernel-based methods, in which data points mapping is computed and the inner products between two data points are denoted as kernel values. Following the existing kernel-based method [17], [21], for the v-th view, we denote the kernel Gram matrix by $\mathbf{H}^{(v)} \in \mathbb{R}^{N\times N}$, and the kernel function by $\kappa(\mathbf{x}, \mathbf{y})$, we have

$$\mathbf{H}_{ij}^{(v)} = \kappa\left(\mathbf{x}_i^{(v)}, \mathbf{x}_j^{(v)}\right) \quad \forall i, j = 1, \ldots, N, \tag{13}$$

where $\kappa\left(\mathbf{x}_i^{(v)}, \mathbf{x}_j^{(v)}\right)$ induces a mapping ϕ. Then, the function $\kappa(\cdot,\cdot)$ can be rewritten as

$$\kappa\left(\mathbf{x}_i^{(v)}, \mathbf{x}_j^{(v)}\right) = \phi\left(\mathbf{x}_i^{(v)}\right)^\top \phi\left(\mathbf{x}_j^{(v)}\right). \tag{14}$$

Given $\Phi\left(\mathbf{X}^{(v)}\right) = \left[\phi\left(\mathbf{x}_1^{(v)}\right), \ldots, \phi\left(\mathbf{x}_N^{(v)}\right)\right]$, and the kernel Gram matrix of the v-th view can be calculated as

$$\mathbf{H}^{(v)} = \Phi\left(\mathbf{X}^{(v)}\right)^\top \Phi\left(\mathbf{X}^{(v)}\right). \tag{15}$$

Commonly, the kernel function includes linear kernel, polynomial kernel, and the Gaussian kernel. For simplicity, in this paper, we define $\kappa\left(\mathbf{x}_i^{(v)}, \mathbf{x}_j^{(v)}\right) = \exp\left(-\frac{\left\|\mathbf{x}_i^{(v)} - \mathbf{x}_j^{(v)}\right\|_2^2}{2\sigma^2}\right)$(Gaussian kernel), where parameter σ denotes the width of the kernel. A larger σ tends to produce a full rank matrix and a smaller σ tends to produce a diagonal matrix. In general, the larger σ is, the fewer clusters the algorithm tends to find.

Although the data points may be nonlinear in original space, the local structure of the data points is useful which can well reveal the intrinsic affinities among data points.

Therefore, according to the existing kernel-based methods, for the v-th view, we have

$$\begin{aligned}\min_{\mathbf{S}^{(v)}} & \sum_{v=1}^{V}\Big(\mathrm{Tr}(\Phi(\mathbf{X}^{(v)})(\mathbf{I}-\mathbf{S}^{(v)})(\mathbf{I}-\mathbf{S}^{(v)})^{\top}\Phi(\mathbf{X}^{(v)^{\top}}) \\ & +\mathrm{Tr}(\mathbf{D}^{(v)^{\top}}\mathbf{S}^{(v)})\Big)+\lambda_1\|\Phi(\mathbf{X}^{(v)})\|_*+\beta\|\mathcal{S}\|_{\circledast} \\ \text{s.t. } & \mathrm{Tr}(\mathbf{S}^{(v)})=c, \mathbf{S}^{(v)^{\top}}=\mathbf{S}^{(v)}, \mathbf{0}\le\mathbf{S}^{(v)}\le\mathbf{1}, \\ & \mathcal{S}=\text{ rotate }\Big(\text{bvfold}([\mathbf{S}^{(1)};\cdots;\mathbf{S}^{(V)}])\Big), \end{aligned} \tag{16}$$

where $\Phi(\cdot)$ represents kernel mapping. $\|\Phi(\mathbf{X}^{(v)})\|_*$ is added to constrain the rank of $\Phi(\mathbf{X}^{(v)})$.

Thus the data points in original nonlinear space are mapped into the high-dimensional kernel space, and the local structure affinities among data points in original space are well preserved [23].

Based on "kernel trick," the kernel Gram matrix can be represented as $\mathbf{H}^{(v)}=\Phi(\mathbf{X}^{(v)})^{\top}\Phi(\mathbf{X}^{(v)})$. Fortunately, because the kernel matrix $\mathbf{H}^{(v)}$ is symmetric positive semi-definite (i.e., $\mathbf{H}^{(v)}\ge 0$), it can be decomposed into $\mathbf{H}^{(v)}=\mathbf{B}^{(v)^{\top}}\mathbf{B}^{(v)}$, that is $\|\mathbf{B}^{(v)}\|_*=\|\Phi(\mathbf{X}^{(v)})\|_*$. This indicates that $\|\mathbf{B}^{(v)}\|_*$ can capture the real affinities among data points.

In addition, if the data points are corrupted by outliers and noise, the computation performance is reduced [52]. In order to effectively process outliers and noise, $\mathbf{H}^{(v)}$ is decomposed into $\mathbf{B}^{(v)^{\top}}\mathbf{B}^{(v)}$ and a sparse noise component $\mathbf{E}^{(v)}$.

Therefore, by considering the above discussions, the proposed RKHSL is given by

$$\begin{aligned}\min_{\mathbf{S}^{(v)},\mathbf{B}^{(v)},\mathbf{E}^{(v)}} & \sum_{v=1}^{V}\Big(\mathrm{Tr}\Big((\mathbf{I}^{(v)}-2\mathbf{S}^{(v)}+\mathbf{S}^{(v)}\mathbf{S}^{(v)^{\top}})\mathbf{B}^{(v)^{\top}}\mathbf{B}^{(v)}\Big) \\ & +\mathrm{Tr}(\mathbf{D}^{(v)^{\top}}\mathbf{S}^{(v)})+\lambda_1\|\mathbf{B}^{(v)}\|_*+\lambda_2\|\mathbf{E}^{(v)}\|_1\Big) \\ & +\beta\|\mathcal{S}\|_{\circledast} \\ \text{s.t. } & \mathrm{Tr}(\mathbf{S}^{(v)})=c, \mathbf{H}^{(v)}=\mathbf{B}^{(v)^{\top}}\mathbf{B}^{(v)}+\mathbf{E}^{(v)}, \\ & \mathbf{S}^{(v)^{\top}}=\mathbf{S}^{(v)}, \mathbf{0}\le\mathbf{S}^{(v)}\le\mathbf{1}, \\ & \mathcal{S}=\text{ rotate }\Big(\text{bvfold}([\mathbf{S}^{(1)};\cdots;\mathbf{S}^{(V)}])\Big). \end{aligned} \tag{17}$$

Combined with the above theory analysis, we propose a kernelized MVGC model. An overall schematic illustration of RKHSL is presented in Figure 1. First, RKHSL learns the affinities between data points and their neighbors by FOS and SOS learning, which can effectively avoid the noises and outliers interference. Second, the tensor is used to learn the TOS among multiple views, so that the consistent information and view-specific information of all views can be efficiently maintained. In addition, a kernel mapping is used to deal with the nonlinear data. Finally, the clustering result is achieved by using spectral clustering.

E. OPTIMIZATION

The (17) can be effectively solved by the augmented Lagrange multiplier-based alternate direction minimization (ALM-ADM) [20], which can iteratively update one variable by fixing the other variables. By introducing the auxiliary variable $\mathcal{A}$, the augmented Lagrange function can be written as

$$\begin{aligned}&\mathcal{L}(\{\mathbf{S}^{(v)},\mathbf{B}^{(v)},\mathbf{E}^{(v)}\}_{v=1}^{V},\mathcal{A}) \\ &=\sum_{v=1}^{V}\Big(\mathrm{Tr}\Big((\mathbf{I}^{(v)}-2\mathbf{S}^{(v)}+\mathbf{S}^{(v)}\mathbf{S}^{(v)^{\top}})\mathbf{B}^{(v)^{\top}}\mathbf{B}^{(v)}\Big) \\ &\quad+\mathrm{Tr}(\mathbf{D}^{(v)^{\top}}\mathbf{S}^{(v)})+\lambda_1\|\mathbf{B}^{(v)}\|_*+\lambda_2\|\mathbf{E}^{(v)}\|_1\Big) \\ &\quad+\beta\|\mathcal{A}\|_{\circledast}+\frac{\mu}{2}\left\|\mathcal{S}-\mathcal{A}+\frac{\mathcal{Y}}{\mu}\right\|_F^2 \\ &\quad+\frac{\rho}{2}\left\|\mathbf{H}^{(v)}-\mathbf{B}^{(v)^{T}}\mathbf{B}^{(v)}-\mathbf{E}^{(v)}+\frac{\mathbf{Y}_2^{(v)}}{\rho}\right\|_F^2 \\ &\text{s.t. } \mathrm{Tr}(\mathbf{S}^{(v)})=c, \mathbf{S}^{(v)^{\top}}=\mathbf{S}^{(v)}, \mathbf{0}\le\mathbf{S}^{(v)}\le\mathbf{1}, \\ &\quad\mathcal{S}=\text{ rotate }\Big(\text{bvfold}([\mathbf{S}^{(1)};\cdots;\mathbf{S}^{(V)}])\Big), \end{aligned} \tag{18}$$

where $\mathcal{Y}$ and $\left\{\mathbf{Y}_2^{(v)}\right\}_{v=1}^{V}$ represent the Lagrangian multipliers, and $\mu>0$, $\rho>0$ represent the penalty parameters. Then, each variable is calculated by fixing the other variables, respectively.

(1) $\mathbf{S}^{(v)}$ *-subproblem:* For updating $\left\{\mathbf{S}^{(v)}\right\}_{v=1}^{V}$ in $\mathcal{S}$, we have the following optimization problem

$$\begin{aligned}\min_{\mathbf{S}^{(v)}} & \Big(\mathrm{Tr}\Big((\mathbf{I}^{(v)}-2\mathbf{S}^{(v)}+\mathbf{S}^{(v)}\mathbf{S}^{(v)^{\top}})\mathbf{B}^{(v)^{\top}}\mathbf{B}^{(v)}\Big) \\ & +\mathrm{Tr}(\mathbf{D}^{(v)^{\top}}\mathbf{S}^{(v)})\Big)+\frac{\mu}{2}\left\|\mathbf{S}^{(v)}-\mathbf{A}^{(v)}+\frac{\mathbf{Y}_1^{(v)}}{\mu}\right\|_F^2 \\ \text{s.t. } & \mathrm{Tr}(\mathbf{S}^{(v)})=c, \mathbf{S}^{(v)^{\top}}=\mathbf{S}^{(v)}, \mathbf{0}\le\mathbf{S}^{(v)}\le\mathbf{1}, \\ & \mathcal{S}=\text{ rotate }\Big(\text{bvfold}([\mathbf{S}^{(1)};\cdots;\mathbf{S}^{(V)}])\Big), \end{aligned} \tag{19}$$

where $\mathbf{A}^{(v)}$ and $\mathbf{Y}_1^{(v)}$ are the v-th slices of the 3-D rotated tensor $\mathcal{A}$ and $\mathcal{Y}$, respectively. Then, (19) can be rewritten as

$$\begin{aligned}\min_{\mathbf{S}^{(v)}} & \frac{1}{2}\left\|\mathbf{S}^{(v)}-\mathbf{R}^{(v)}\right\|_F^2 \\ \text{s.t. } & \mathrm{Tr}(\mathbf{S}^{(v)})=c, \mathbf{S}^{(v)^{\top}}=\mathbf{S}^{(v)}, \mathbf{0}\le\mathbf{S}^{(v)}\le\mathbf{1}, \end{aligned} \tag{20}$$

where

$$\begin{aligned}\mathbf{R}^{(v)}=&\Big(2\mathbf{B}^{(v)^{\top}}\mathbf{B}^{(v)}+\mu\mathbf{I}\Big)^{-1}\Big(\mu\mathbf{A}^{(v)}-\mathbf{Y}_1^{(v)} \\ &+2\mathbf{B}^{(v)^{\top}}\mathbf{B}^{(v)}-\mathbf{D}^{(v)^{\top}}\Big). \end{aligned} \tag{21}$$

The above discussed problem can be well solved according to Theorem 2.

Theorem 2 [20]: For a symmetric affinity matrix $\mathbf{S}\in\mathbb{R}^{N\times N}$*, the SVD of* $\mathbf{S}$ *is denoted as* $\mathbf{R}=\mathbf{U}\,\mathrm{Diag}(\boldsymbol{\zeta})\mathbf{U}^{\top}$*. The following problem*

$$\min_{\mathbf{S}}\frac{1}{2}\|\mathbf{S}-\mathbf{R}\|_F^2 \quad s.t.\ \mathrm{Tr}(\mathbf{S})=c, \mathbf{S}^{\top}=\mathbf{S}, \mathbf{0}\le\mathbf{S}\le\mathbf{1} \tag{22}$$

has the optimal solution given by $\mathbf{S}^* = \mathbf{U}\,\mathrm{Diag}\,(\boldsymbol{\rho}^*)\,\mathbf{U}^\top$, *where* $\boldsymbol{\rho}^*$ *is the solution to*

$$\min_{\boldsymbol{\rho}} \frac{1}{2}\|\boldsymbol{\rho}-\boldsymbol{\zeta}\|_2^2, \quad s.t.\ 0 \le \boldsymbol{\rho} \le 1, \boldsymbol{\rho}^\top 1 = c. \tag{23}$$

Finally, (23) can be effectively solved with an alternate iterative algorithm as [53].

(2) $\mathbf{B}^{(v)}$*-subproblem:* For updating $\mathbf{B}^{(v)}$, the optimization problem is denoted as

$$\min_{\mathbf{B}^{(v)}} \mathrm{Tr}\left((\mathbf{I}^{(v)}-2\mathbf{S}^{(v)}+\mathbf{S}^{(v)}\mathbf{S}^{(v)\top})\mathbf{B}^{(v)\top}\mathbf{B}^{(v)}\right)+\lambda_1\|\mathbf{B}^{(v)}\|_* + \frac{\rho}{2}\left\|\mathbf{H}^{(v)}-\mathbf{B}^{(v)\top}\mathbf{B}^{(v)}-\mathbf{E}^{(v)}+\frac{\mathbf{Y}_2^{(v)}}{\rho}\right\|_F^2. \tag{24}$$

Let $\mathbf{W}^{(v)} = \mathbf{H}^{(v)} - \mathbf{E}^{(v)} + \frac{\mathbf{Y}_2^{(v)}}{\rho}$, the optimization problem for (24) can be reformulated as

$$\min_{\mathbf{B}^{(v)}} \lambda_1\|\mathbf{B}^{(v)}\|_* + \frac{\rho}{2}\left\|\mathbf{B}^{(v)\top}\mathbf{B}^{(v)}-\tilde{\mathbf{W}}^{(v)}\right\|_F^2, \tag{25}$$

where $\tilde{\mathbf{W}}^{(v)} = \mathbf{W}^{(v)} - \frac{1}{\rho}\left(\mathbf{I}^{(v)}-2\mathbf{S}^{(v)}+\mathbf{S}^{(v)}\mathbf{S}^{(v)\top}\right)$. Fortunately, such a problem can be effectively solved according to Theorem 3.

Theorem 3 [23]: Given $\mathbf{A} \ge \mathbf{0}$, $\mathbf{A} = \mathbf{U}\Sigma\mathbf{U}^\top$ *represents the singular value decomposition of* $\mathbf{A}$. *Then*

$$\min_{\mathbf{G}} \frac{\kappa}{2}\left\|\mathbf{A}-\mathbf{G}^\top\mathbf{G}\right\|_F^2 + \tau\|\mathbf{G}\|_* = \sum_{i=1}^{n}\left(\frac{\kappa}{2}\left(\sigma_i-\gamma_i^{*2}\right)^2+\tau\gamma_i^*\right). \tag{26}$$

A minimizer $\mathbf{G}^*$ *of* (26) *is given by*

$$\mathbf{G}^* = \Gamma^*\mathbf{U}^T \tag{27}$$

with $\Gamma^* \in \mathbf{Q}_+^n$, $\gamma_i^* \in \left\{\alpha \in \mathbb{R}_+ \mid p_{\sigma_i,\tau/2\kappa}(\alpha)=0\right\} \cup \{0\}$, *in which* $p_{a,b}$ *represents the depressed cubic* $p_{a,b}(x) = x^3 - ax + b$. $\mathbf{Q}_+^n$ *represents the set of* $n \times n$ *diagonal matrix with non-negative entries.*

(3) $\mathbf{E}^{(v)}$*-subproblem:* For updating $\mathbf{E}^{(v)}$, the optimization problem is given by

$$\min_{\mathbf{E}^{(v)}} \lambda_2\|\mathbf{E}^{(v)}\|_1 + \frac{\rho}{2}\left\|\mathbf{H}^{(v)}-\mathbf{B}^{(v)\top}\mathbf{B}^{(v)}-\mathbf{E}^{(v)}+\frac{\mathbf{Y}_2^{(v)}}{\rho}\right\|_F^2. \tag{28}$$

Let $\mathbf{O}^{(v)} = \mathbf{H}^{(v)} - \mathbf{B}^{(v)\top}\mathbf{B}^{(v)} + \frac{\mathbf{Y}_2^{(v)}}{\rho}$, the problem for (28) can be solved according to

$$\min_{\mathbf{E}_i^{(v)}} \frac{\rho}{2}\left\|\mathbf{E}_i^{(v)}-\mathbf{O}_i^{(v)}\right\|_2^2 + \lambda_2\left\|\mathbf{E}_i^{(v)}\right\|_1, \tag{29}$$

where $\mathbf{E}_i^{(v)}$ and $\mathbf{O}_i^{(v)}$ are the i-th column of $\mathbf{E}^{(v)}$ and $\mathbf{O}^{(v)}$, respectively. The solution for this problem is given by

$$\mathbf{E}_{ij}^{(v)*} = \mathrm{sign}\left(\mathbf{O}_{ij}^{(v)}\right)\left(\mathrm{abs}\left(\mathbf{O}_{ij}^{(v)}\right)-\frac{\lambda_2}{\rho}\right)_+ = \begin{cases}\mathbf{O}_{ij}^{(v)}-\frac{\lambda_2}{\rho}, & \text{if } \mathbf{O}_{ij}^{(v)} > \frac{\lambda_2}{\rho}\\ \mathbf{O}_{ij}^{(v)}+\frac{\lambda_2}{\rho}, & \text{if } \mathbf{O}_{ij}^{(v)} < -\frac{\lambda_2}{\rho}\\ 0, & \text{otherwise}.\end{cases} \tag{30}$$

(4) $\boldsymbol{\mathcal{A}}$*-subproblem:* For updating $\boldsymbol{\mathcal{A}}$, the optimization problem is given by

$$\min_{\boldsymbol{\mathcal{A}}} \beta\|\boldsymbol{\mathcal{A}}\|_\circledast + \frac{\mu}{2}\left\|\boldsymbol{\mathcal{A}}-\left(\boldsymbol{\mathcal{S}}+\frac{\boldsymbol{\mathcal{Y}}}{\mu}\right)\right\|_F^2, \tag{31}$$

which can be considered as a t-TNN minimization problem. Given $\boldsymbol{\mathcal{R}} = \boldsymbol{\mathcal{S}} + \frac{\boldsymbol{\mathcal{Y}}}{\mu}$, (31) can be effectively solved by Theorem 4.

Theorem 4 [15]: The 3-order tensors $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{N_1\times N_2\times N_3}$, $\boldsymbol{\mathcal{C}} \in \mathbb{R}^{N_1\times N_2\times N_3}$, *and a scalar* $\tau > 0$ *are given, the following problem*

$$\min_{\boldsymbol{\mathcal{A}}} \tau\|\boldsymbol{\mathcal{A}}\|_\circledast + \frac{1}{2}\|\boldsymbol{\mathcal{A}}-\boldsymbol{\mathcal{C}}\|_F^2 \tag{32}$$

can be effectively solved by the tensor tubal-shrinkage operator, i.e.,

$$\boldsymbol{\mathcal{A}} = \mathcal{F}_{N_3\tau}(\boldsymbol{\mathcal{C}}) = \boldsymbol{\mathcal{U}} * \mathcal{F}_{N_3\tau}(\boldsymbol{\mathcal{M}}) * \boldsymbol{\mathcal{V}}^\top, \tag{33}$$

where $\boldsymbol{\mathcal{C}} = \boldsymbol{\mathcal{U}} * \boldsymbol{\mathcal{M}} * \boldsymbol{\mathcal{V}}^\top$, $\mathcal{F}_{N_3\tau} = \boldsymbol{\mathcal{M}} * \boldsymbol{\mathcal{Q}}$. *The tensor* $\boldsymbol{\mathcal{Q}} \in \mathbb{R}^{N_1\times N_2\times N_3}$ *is f-diagonal, and the diagonal element of* $\boldsymbol{\mathcal{Q}} \in \mathbb{R}^{N_1\times N_2\times N_3}$ *is represented by* $\boldsymbol{\mathcal{Q}}_f(i,i,j) = \left(1-\frac{N_3\tau}{\boldsymbol{\mathcal{M}}(i,i,j)}\right)_+$.

(5) *ADMM variables:* The Lagrange multiplier $\boldsymbol{\mathcal{Y}}$ can be updated by

$$\begin{aligned}\boldsymbol{\mathcal{Y}} &= \boldsymbol{\mathcal{Y}} + \mu(\boldsymbol{\mathcal{A}}-\boldsymbol{\mathcal{S}}),\\ \mu &= \min\left(\nu\mu, \mu_{\max}\right),\\ \rho &= \min\left(\nu\rho, \rho_{\max}\right),\end{aligned} \tag{34}$$

where μ and ρ represent the scalars involved in ADMM.

In each iterator, the algorithm convergence is checked via

$$\text{error} = \|\boldsymbol{\mathcal{A}}-\boldsymbol{\mathcal{S}}\|_\infty < \epsilon, \tag{35}$$

where $\epsilon = 10^{-7}$ is a threshold. The tensor $\boldsymbol{\mathcal{S}}$ with size $N \times V \times N$ can be rotated to $\boldsymbol{\mathcal{S}}^* = \mathrm{irotate}\,(\boldsymbol{\mathcal{S}})$ with size $N \times N \times V$. Then, the optimal affinity graph $\overline{\mathbf{S}}$ can be computed by averaging all the frontal slices of $\boldsymbol{\mathcal{S}}^*$, *i.e.*,

$$\overline{\mathbf{S}} = \frac{1}{V}\sum_{v=1}^{V}\boldsymbol{\mathcal{S}}^*(:,:,v). \tag{36}$$

Subsequently, by using $\overline{\mathbf{S}}$ as input, the final clustering assignments are achieved by using spectral clustering. The optimization algorithm of RKHSL is described in Algorithm 1.

F. COMPLEXITY ANALYSIS

The computation complexity of Algorithm 1 is as follows: Updating $\mathbf{S}^{(v)}$ has the computation complexity of $\mathcal{O}\left(VN^3\right)$. For updating $\mathbf{B}^{(v)}$, the computational complexity is $\mathcal{O}\left(VN^3\right)$. The updating $\mathbf{E}^{(v)}$ is element-wise operation, which has the computational complexity of $\mathcal{O}\left(VN^2\right)$. Updating $\boldsymbol{\mathcal{A}}$ need to

Algorithm 1 Optimization for RKHSL.

Input: Multi-view matrices $\mathbf{X} = [\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(V)}]$, parameters λ_1, λ_2 and β.

Output: The clustering results.

1: Initialize $\mathcal{A} = \mathcal{S}$, $\epsilon = 10^{-7}$, $\mu = 10^{-4}$, $\rho = 10^{-4}$, and $maxIter = 50$.
2: **while** convergence criterion $\epsilon < 10^{-7}$ is not satisfied **do**
3: Update each affinity graph $\mathbf{S}^{(v)}$ via (19);
4: Update $\mathbf{B}^{(v)}$ via (24);
5: Update $\mathbf{E}^{(v)}$ via (28);
6: Update graph tensor $\mathcal{A}$ via (31);
7: Update ADMM involved variables via (34);
8: **end while**
9: Perform spectral clustering on affinity graph $\overline{\mathbf{S}}$.

TABLE 2. Summary of the benchmark datasets.

Dataset	Images	Objective	Clusters
BBCSport	544	News article	5
Flowers	1360	Flower	17
UCI digits	2000	Digit	10
COIL-20	1440	Generic object	20
Scene-15	4485	Scene	15

compute the tensor FFT and inverse FFT along the third dimension of $\mathcal{S} \in \mathbb{R}^{N\times V\times N}$, which has the computation complexity of $\mathcal{O}\left(VN^2\log(N)\right)$; moreover, the SVD of all frontal slices of $\mathcal{S}$ in the Fourier domain has the computation complexity of $\mathcal{O}\left(V^2N^2\right)$. Therefore, updating $\mathcal{A}$ has the computation complexity of $\mathcal{O}\left(VN^2\log(N) + V^2N^2\right)$. For the auxiliary variables involved in ADMM, the computation complexity is $\mathcal{O}(V)$ in each iteration. Therefore, the complexity of RKHSL is $\mathcal{O}(t((V^2+V)N^2+VN^2\log(N)+2VN^3+V))$, where t denotes the iterations. The spectral clustering has the computation complexity of $\mathcal{O}(N^3)$. In practice, we have $t \ll N$ and $V \ll N$. Thus Algorithm 1 has the computation complexity of $\mathcal{O}(N^3)$.

V. EXPERIMENT

All these experiments are performed on an Intel Core i5 (2.9 GHz) CPU, with a Windows 10(x64) operating system, 16 GB memory, and the MATLAB 2019b simulation software.

A. DATASETS

We adopt five real public datasets in our experiments and summarize the related information in Table 2, which include text, flower, digit, generic object, and scene. Then we briefly introduce these datasets as follows.

BBCSport[1] includes 544 sports news refer to five thematic areas on the BBC website. In our experiment, similar to [27], it contains two views with dimensions of 3183 as view 1 and 3203 as view 2.

[1]http://mlg.ucd.ie/datasets/segment.html

TABLE 3. Parameter settings of RKHSL.

Parameter	BBCSport	Flowers	UCI digits	COIL-20	Scene-15
λ_1	0.01	1000	1	10	1
λ_2	10000	1000	100	10	1
β	10000	1000	100	10^4	1000

Flowers[2] contains 1360 flower samples with 17 categories. In our experiment, similar to [27], three features are extracted as three views, which are 1360d color feature as view 1, 1360d texture feature as view 2, and 1360d shape feature as view 3.

UCI digits[3] includes 2, 000 digit images of handwritten numbers (0-9) corresponding to 10 categories. In our experiment, similar to [54], three different kinds of features are extracted to represent these digit images, which are Fourier coefficients feature, pixel averages feature, and morphological feature.

COIL-20[4] consists of 1, 440 images corresponding to 20 categories. In our experiment, similar to [34], three different types of features are extracted to represent these images, including intensity, LBP, and Gabor features.

Scene-15 [55] includes 15 categories of indoor and outdoor natural environment scenes, including industrial, kitchen, store, etc., with a total of 4, 485 images. Similar to [16], three different types of features are extracted to represent these scene images, including PHOW (pyramid histogram of words), LBP, and CENTRIST (census transform histogram).

B. BASELINES AND EVALUATION METRICS

The proposed method is compared with two classical single-view clustering methods and six advanced multi-view clustering methods.

- SPCbest [56]: Best spectral clustering (SPCbest) achieves the best clustering result by using standard SPC.
- LRRbest [57]: Best low-rank representation (LRRbest) achieves the best clustering result by adopting a low-rank constraint.
- RMSC [54]: Robust MVSC (RMSC) achieves the MVC results by recovering a shared low-rank transition probability matrix of the Markov chain.
- DiMSC [58]: Diversity-induced multi-view subspace clustering (DiMSC) learns the complementary information of multiple views by using HSIC as the diversity term.
- MCLES [24]: MVC in latent embedding space (MCLES) achieves the MVC by learning a latent embedding representation from all views.
- LTMSC [34]: Low-rank tensor constrained MSC (LTMSC) learns the optimal self-representation matrix

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

TABLE 4. Performance comparison of different clustering methods on the BBCSport dataset.

Method	ACC	NMI	F-score	Precision	AR	Recall
SPCbest [56]	0.036(0.002)	0.344(0.012)	0.249(0.001)	0.369(0.020)	0.381(0.005)	0.297(0.007)
LRRbest [57]	0.836(0.001)	0.698(0.002)	0.776(0.001)	0.768(0.001)	0.705(0.001)	0.784(0.001)
RMSC [54]	0.826(0.001)	0.666(0.001)	0.719(0.001)	0.766(0.001)	0.637(0.001)	0.677(0.001)
DiMSC [58]	0.922(0.000)	0.785(0.000)	0.858(0.000)	0.846(0.000)	0.813(0.000)	0.872(0.000)
MCLES [24]	0.921(0.000)	0.802(0.000)	0.845(0.000)	0.827(0.000)	0.795(0.000)	0.865(0.000)
LTMSC [34]	0.476(0.030)	0.230(0.018)	0.432(0.010)	0.335(0.020)	0.178(0.031)	0.335(0.020)
t-SVD-MSC [15]	0.949(0.000)	0.894(0.000)	0.938(0.000)	0.935(0.000)	0.918(0.000)	0.940(0.000)
ETLMSC [16]	0.934(0.000)	0.827(0.000)	0.877(0.000)	0.901(0.000)	0.840(0.000)	0.855(0.000)
HOSL	0.974(0.000)	0.938(0.000)	0.951(0.000)	0.974(0.000)	0.936(0.000)	0.928(0.000)
RKHSL	**1.000(0.000)**	**1.000(0.000)**	**1.000(0.000)**	**1.000(0.000)**	**1.000(0.000)**	**1.000(0.000)**

TABLE 5. Performance comparison of different clustering methods on the flowers dataset.

Method	ACC	NMI	F-score	Precision	AR	Recall
SPCbest [56]	0.070(0.003)	0.027(0.002)	0.110(0.002)	0.932(0.140)	0.001(0.000)	0.058(0.000)
LRRbest [57]	0.396(0.009)	0.419(0.006)	0.284(0.003)	0.279(0.004)	0.239(0.003)	0.290(0.003)
RMSC [54]	0.385(0.016)	0.396(0.001)	0.249(0.011)	0.234(0.012)	0.231(0.019)	0.256(0.010)
DiMSC [58]	0.434(0.014)	0.442(0.011)	0.310 (0.008)	0.302(0.007)	0.266(0.009)	0.318(0.010)
MCLES [24]	0.469(0.000)	0.516(0.000)	0.390(0.000)	0.342(0.000)	0.337(0.000)	0.462(0.000)
t-SVD-MSC [15]	0.836(0.005)	0.852(0.002)	0.780(0.002)	0.772(0.002)	0.766(0.002)	0.789(0.002)
ETLMSC [16]	0.811(0.066)	0.874(0.025)	0.778(0.054)	0.748(0.064)	0.763(0.057)	0.810(0.041)
HOSL	0.881(0.038)	0.918(0.014)	0.856(0.033)	0.841(0.044)	0.849(0.035)	0.875(0.022)
RKHSL	**0.975(0.019)**	**0.968(0.009)**	**0.956(0.018)**	**0.958(0.013)**	**0.954(0.019)**	**0.955(0.028)**

TABLE 6. Performance comparison of different clustering methods on the UCI digits dataset.

Method	ACC	NMI	F-score	Precision	AR	Recall
SPCbest [56]	0.731(0.034)	0.642(0.021)	0.591(0.029)	0.582(0.030)	0.545(0.033)	0.601(0.030)
LRRbest [57]	0.720(0.047)	0.743(0.021)	0.690(0.028)	0.734(0.021)	0.654(0.032)	0.652(0.033)
RMSC [54]	0.915(0.036)	0.822(0.026)	0.811(0.049)	0.797(0.065)	0.789(0.055)	0.826(0.031)
DiMSC [58]	0.867(0.001)	0.782(0.002)	0.772(0.002)	0.769(0.002)	0.747(0.002)	0.775(0.002)
MCLES [24]	0.941(0.004)	0.891(0.008)	0.889(0.008)	0.885(0.008)	0.877(0.009)	0.894(0.007)
LTMSC [34]	0.792(0.009)	0.762(0.009)	0.737(0.013)	0.724(0.012)	0.707(0.014)	0.749(0.013)
t-SVD-MSC [15]	0.966(0.001)	0.934(0.001)	0.935(0.001)	0.933(0.001)	0.928(0.001)	0.936(0.001)
ETLMSC [16]	0.941(0.023)	0.970(0.013)	0.936(0.027)	0.935(0.031)	0.933(0.029)	0.938(0.024)
HOSL	0.983(0.000)	0.959(0.000)	0.967(0.000)	0.965(0.000)	0.962(0.000)	0.966(0.000)
RKHSL	**0.999(0.000)**	**0.996(0.000)**	**0.997(0.000)**	**0.997(0.000)**	**0.997(0.000)**	**0.997(0.000)**

by using a low-rank tensor constraint to improve clustering performance.

- t-SVD-MSC [15]: t-SVD based MSC (t-SVD-MSC) learns the optimal self-representation matrix by using the t-SVD based on tensor rotation to improve clustering performance.
- ETLMSC [16]: Essential tensor learning for MVSC (ETLMSC) is a novel tensor-based spectral clustering method by using the multi-view transition probability matrices of the Markov chain to excavate the high-order relations of multiple views.
- HOSL: MVC based on high-order similarity learning (HOSL) is presented in (12) in section IV.

For evaluating the comparison methods mentioned above, five benchmark metrics are used, including Accuracy (ACC), Normalized Mutual Information (NMI), F-score, Precision, adjusted rand index (AR), and Recall [59]. For these metrics, the higher values indicate better performance. To avoid the random value in our experiments, each method is performed 20 times and the mean and standard deviation are reported. To comprehensively evaluate our work, these methods include single view clustering, MVC, and tensor-based clustering.

C. EXPERIMENTAL RESULTS

Abundant MVC experiments are performed for evaluating our work and the above comparison methods. The experimental results are shown in Tables 4-8, where the numbers in the parentheses refer to the standard deviations.

From these comparison results, the following results can be observed:

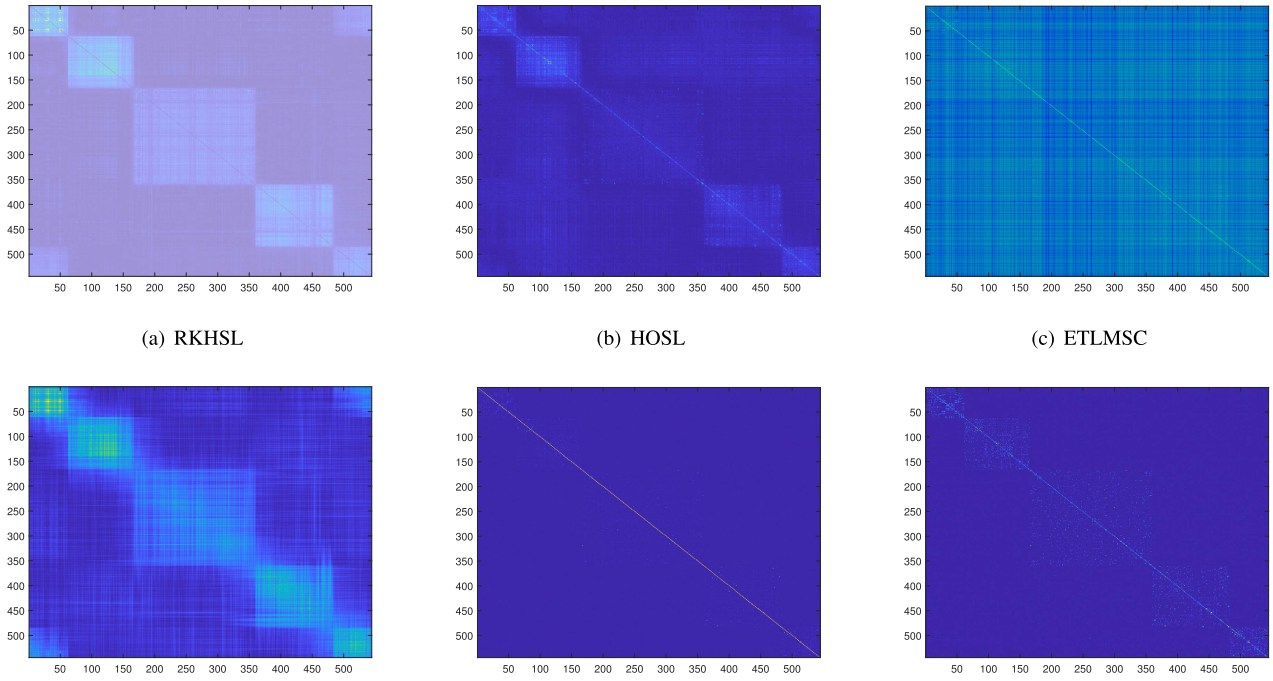
- The single-view clustering, SPCbest and LRRbest, have obtained better performance. But in the main, MVC performs better than single view clustering. This verifies that the complementary information included in multiple views can greatly improve the clustering performance.
- Overall, as shown in Tables 4-8, the RKHSL achieves the best performance on all the datasets under

TABLE 7. Performance comparison of different clustering methods on the COIL-20 dataset.

Method	ACC	NMI	F-score	Precision	AR	Recall
SPCbest [56]	0.672(0.063)	0.806(0.008)	0.640(0.017)	0.596(0.021)	0.619(0.018)	0.692(0.013)
LRRbest [57]	0.761(0.003)	0.829(0.006)	0.734(0.006)	0.717(0.003)	0.720(0.020)	0.751(0.002)
RMSC [54]	0.685(0.045)	0.800(0.017)	0.656(0.042)	0.620(0.057)	0.637(0.044)	0.698(0.026)
DiMSC [58]	0.778(0.022)	0.846(0.002)	0.745(0.005)	0.739(0.007)	0.732(0.005)	0.751(0.003)
MCLES [24]	0.706(0.026)	0.740(0.019)	0.553(0.029)	0.505(0.036)	0.521(0.032)	0.611(0.021)
LTMSC [34]	0.804(0.011)	0.860(0.002)	0.760(0.007)	0.741(0.009)	0.748(0.004)	0.776(0.006)
t-SVD-MSC [15]	0.830(0.000)	0.884(0.005)	0.800(0.004)	0.785(0.007)	0.786(0.003)	0.808(0.001)
ETLMSC [16]	0.873(0.065)	0.945(0.030)	0.872(0.060)	0.840(0.068)	0.865(0.063)	0.907(0.055)
HOSL	0.956(0.035)	0.968(0.015)	0.939(0.033)	0.933(0.039)	0.936(0.035)	0.946(0.029)
RKHSL	**0.990(0.022)**	**0.995(0.010)**	**0.989(0.026)**	**0.990(0.019)**	**0.989(0.024)**	**0.987(0.028)**

TABLE 8. Performance comparison of different clustering methods on the Scene-15 dataset.

Method	ACC	NMI	F-score	Precision	AR	Recall
SPCbest [56]	0.437(0.015)	0.421(0.010)	0.321(0.022)	0.314(0.016)	0.270(0.010)	0.329(0.020)
LRRbest [57]	0.445(0.013)	0.426(0.018)	0.324(0.010)	0.316(0.015)	0.272(0.015)	0.333(0.015)
RMSC [54]	0.507(0.017)	0.564(0.023)	0.437(0.019)	0.425(0.021)	0.394(0.025)	0.450(0.024)
DiMSC [58]	0.300(0.010)	0.269(0.009)	0.181(0.012)	0.173(0.016)	0.117(0.012)	0.190(0.010)
LTMSC [34]	0.574(0.009)	0.571(0.011)	0.465(0.007)	0.452(0.003)	0.424(0.010)	0.479(0.008)
t-SVD-MSC [15]	0.812(0.007)	0.858(0.007)	0.788(0.001)	0.743(0.006)	0.771(0.003)	0.839(0.003)
ETLMSC [16]	0.865(0.030)	0.879(0.006)	0.828(0.018)	0.825(0.020)	0.816(0.019)	0.832(0.016)
HOSL	**0.907(0.034)**	0.916(0.007)	0.864(0.020)	0.889(0.022)	0.854(0.022)	0.857(0.019)
RKHSL	0.893(0.002)	**0.942(0.002)**	**0.900(0.003)**	**0.906(0.005)**	**0.893(0.004)**	**0.895(0.002)**



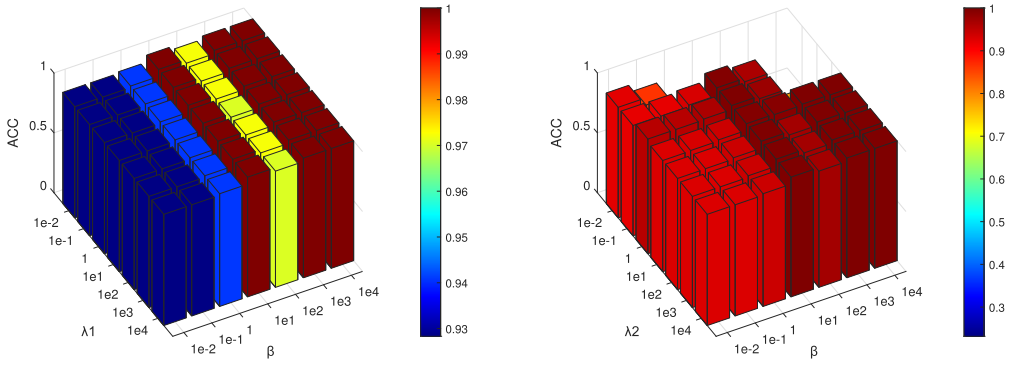
(a) RKHSL (b) HOSL (c) ETLMSC

(d) t-SVD-MSC (e) LTMSC (f) MCLES

FIGURE 3. Visualizations for affinity graphs learned from the different methods on the BBCSport dataset. As presented, our RKHSL produces much more clear structures.

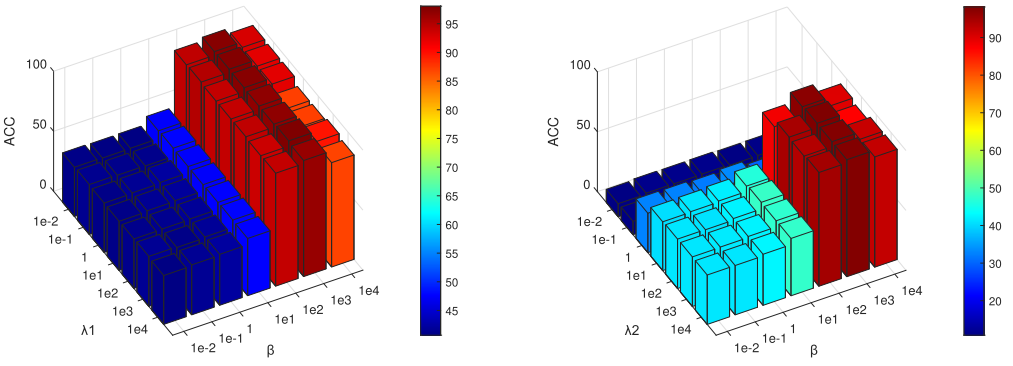
six evaluation metrics. For example, for ACC, from Table 4 and Table 5, the improvements of RKHSL over the best competitor HOSL are about 3% and 9% on BBCSport and Flowers datasets, respectively.

- Compared with RMSC, DiMSC, and MCLES, our proposed RKHSL gains a remarkable improvement. The primary reason is that RMSC, DiMSC, and MCLES only acquire the common information among multiple views,



(a) The ACC versus λ_1 and β (b) The ACC versus λ_2 and β

FIGURE 4. The ACC of RKHSL on the BBCSport dataset with different parameter settings.



(a) The ACC versus λ_1 and β (b) The ACC versus λ_2 and β

FIGURE 5. The ACC of RKHSL on the Flowers dataset with different parameter settings.

while our proposed RKHSL learns the 3-order similarity among multiple views which can effectively excavate each view-specific information and multi-view consist information.

- Compared with ETLMSC and t-SVD-MSC, our proposed RKHSL gains significant improvement on all datasets, especially on the difficult Scene-15 datasets. We can easily observe that RKHSL improves 3% and 8% in terms of ACC than ETLMSC and t-SVD-MSC on Scene-15 datasets, which shows the effectiveness of FOS and SOS simultaneous learning.
- Compared with HOSL, our proposed RKHSL performs well. The primary reason is that we introduce the kernel mapping to further learn the nonlinear data, both the linear affinities in the mapping kernel space and the locally structure affinities in the original space can be captured simultaneously. Therefore, the RKHSL obtains the perfect results.

D. VISUAL VERIFICATION OF BLOCK DIAGONAL REPRESENTATION

The importance of affinity graph for multi-view clustering has been discussed earlier. Therefore, we illustrate the affinity graphs of the compared methods on the BBCSport dataset in Figure 3. It is observed that the affinity graph satisfies the block diagonal property with exact c connected blocks, where each block corresponds one-to-one with one cluster of the data points. For example, from Figure 3, the affinity graph learned from RKHSL has exact 5 block diagonal components

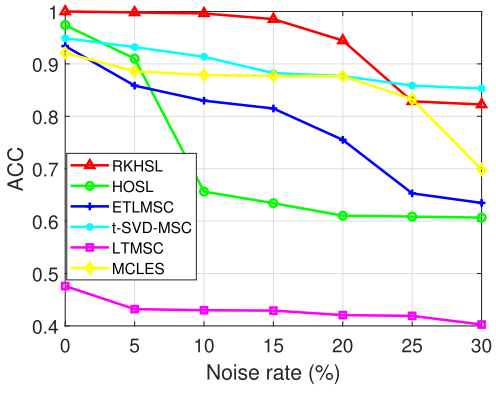
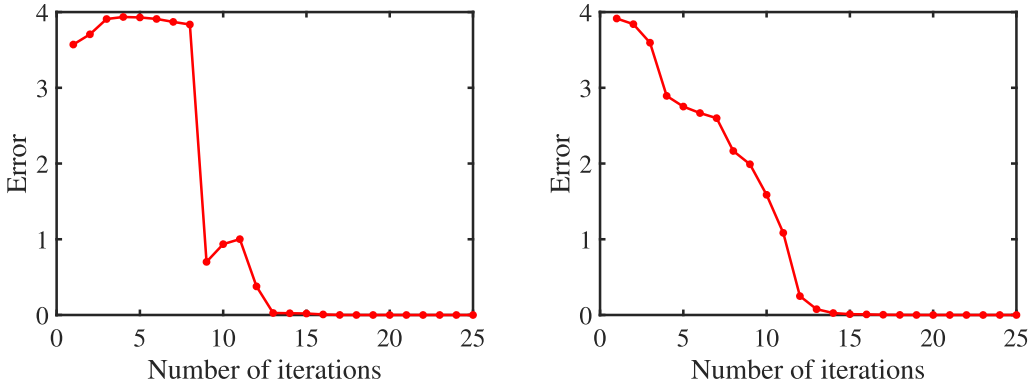


FIGURE 6. Plots of clustering accuracy (ACC) on BBCSport dataset with missing data noise.



(a) BBCSport (b) Flowers

FIGURE 7. Convergence curves of RKHSL on the BBCSport dataset and Flowers dataset.

clearly because the BBCSport dataset contains 5 clusters. The inter-cluster affinities are all zeros, but the intra-cluster affinities are not-zeros. This indicates that the data points placed in the same cluster have greater affinity than data points placed in different clusters. Considering the cluster diameter and cluster spacing, the maximum distances between data points within clusters are less than the minimum distances between data points in different clusters. And the cluster separation is ideal if the diameter of each cluster is smaller than each of its cluster spacing.

Therefore, the above visual results demonstrate that RKHSL has admirable clustering performance; moreover, they further verifies that RKHSL can well explore the nonlinear affinities among data points.

E. PARAMETER ANALYSIS

Table 3 gives the optimal values of λ_1, λ_2 and β involved in our algorithm. The grid search method is used to show the influences of the three hyper-parameters. Taking the BBCSport and Flowers datasets for example, we present the ACC of our proposed RKHSL method in Figure 4 and Figure 5, respectively. We first fix $\lambda_2 = 10000$ on BBCSport dataset and $\lambda_2 = 1000$ on Flowers dataset, and then turn λ_1 and β in $[10^{-2}, \cdots, 10^4]$. From Figure 4(a) and Figure 5(a), we can observe that our algorithm is not sensitive to λ_1. Besides, the results suggest that when β varies in $[10^2, \cdots, 10^4]$, clustering performance is relatively stable.

Next, we fix $\lambda_1 = 0.01$ on BBCSport dataset and $\lambda_1 = 1000$ on Flowers dataset, and then turn λ_2 and β in $[10^{-2}, \cdots, 10^4]$. From Figure 4(b) and Figure 5(b), we can observe that the RKHSL can achieve the desired effect when λ_2 varies in $[10^1, \cdots, 10^4]$. In summary, when these hyper-parameters vary in a relatively large interval, our algorithm is insensitive. Besides, these hyper-parameters are insensitive to different datasets. On all the five datasets, we use the same set of hyper-parameters to achieve the comparative results shown in Tables 4-8.

F. ROBUSTNESS EXPERIMENT

In this part, we test the robustness of RKHSL to non-Gaussian noise on the BBCSport dataset. In our experiments, we randomly select some samples and set them as zero to simulate the missing noise data. All results are presented in Figure 6. We can clearly see that RKHSL has good robustness. When the data is corrupted, RKHSL can fully excavate the potential affinities among data points and different views to ensure the validity of the learned consensus affinity graph.

G. CONVERGENCE ANALYSIS

The convergence of our algorithm is analyzed in this part. The error of our algorithm in each iteration is shown in Figure 7 on the BBCSport dataset and Flowers dataset. In our algorithm, the error is defined as the change in each iteration: error $= \|\mathcal{A} - \mathcal{S}\|_\infty$. According to Figure 7, the error reduces with the increasing of iterations which indexes that the optimization algorithm is convergent. The error value decreases quickly with the increase of iterations and converges within 50 iterations, and similar phenomena can be observed for other datasets.

VI. CONCLUSION

In this paper, we put forward a robust kernelized MVC based on high-order similarity learning (RKHSL). Based on FOS and SOS learning, we explore the local structure affinities among data points in original space and excavate the nonlinear affinities among data points in the high-dimensional kernel spaces. Afterwards, the TOS from multiple input affinity graphs of multiple views is captured in a 3-order tensor with a low-rank nuclear norm constraint. By t-SVD and tensor rotation operation, the RKHSL can be highly optimized. We evaluate the performance of our work on five open datasets in multiple applications, and it achieves significant improvement compared with the current state-of-the-art methods under six benchmark metrics.

In future work, we would like to concentrate on the multiple kernels learning for MVC, which can effectively utilize the high-order similarity among data points for clustering.

REFERENCES

[1] Y. Lin, Y. Tu, Z. Dou, L. Chen, and S. Mao, "Contour Stella image and deep learning for signal recognition in the physical layer," *IEEE Trans. Cognit. Commun. Netw.*, vol. 7, no. 1, pp. 34–46, Mar. 2021.

[2] Y. Yang and H. Wang, "Multi-view clustering: A survey," *Bid Data Mining Anal.*, vol. 1, no. 2, pp. 83–107, Jun. 2018.

[3] M. Saha, "A graph based approach to multiview clustering," in *Proc. Int. Conf. Pattern Recognit. Mach. Intell.* Berlin, Germany: Springer, 2013, pp. 128–133.

[4] 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, Mar. 2013.

[5] J. Zhao, X. Xie, X. Xu, and S. Sun, "Multi-view learning overview: Recent progress and new challenges," *Inf. Fusion*, vol. 38, pp. 43–54, Nov. 2017.

[6] H. Wang, Y. Yang, and B. Liu, "GMC: Graph-based multi-view clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 6, pp. 1116–1129, May 2020.

[7] R. Vidal and P. Favaro, "Low rank subspace clustering," *Pattern Recognit. Lett.*, vol. 43, no. 1, pp. 47–61, 2014.

[8] J. Ma, Y. Zhang, and L. Zhang, "Discriminative subspace matrix factorization for multiview data clustering," *Pattern Recognit.*, vol. 111, Mar. 2021, Art. no. 107676.

[9] F. Nie, X. Wang, and H. Huang, "Clustering and projected clustering with adaptive neighbors," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2014, pp. 977–986.

[10] E. Elhamifar and R. Vidal, "Clustering disjoint subspaces via sparse representation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2010, pp. 1926–1929.

[11] Z. Kang, H. Xu, B. Wang, H. Zhu, and Z. Xu, "Clustering with similarity preserving," *Neurocomputing*, vol. 365, pp. 211–218, Nov. 2019.

[12] W. Zhuge, F. Nie, C. Hou, and D. Yi, "Unsupervised single and multiple views feature extraction with structured graph," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 10, pp. 2347–2359, Oct. 2017.

[13] W. Liang, S. Zhou, J. Xiong, X. Liu, S. Wang, E. Zhu, Z. Cai, and X. Xu, "Multi-view spectral clustering with high-order optimal neighborhood Laplacian matrix," *IEEE Trans. Knowl. Data Eng.*, early access, Sep. 18, 2020, doi: 10.1109/TKDE.2020.3025100.

[14] Y. Tian, X. Yu, B. Fan, F. Wu, H. Heijnen, and V. Balntas, "SOSNet: Second order similarity regularization for local descriptor learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11016–11025.

[15] Y. Xie, D. Tao, W. Zhang, 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.

[16] 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.

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

[18] B. Wang, Y. Hu, J. Gao, Y. Sun, and B. Yin, "Laplacian LRR on product Grassmann manifolds for human activity clustering in multicamera video surveillance," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 3, pp. 554–566, Mar. 2017.

[19] B. Wang, Y. Hu, J. Gao, Y. Sun, and B. Yin, "Partial sum minimization of singular values representation on Grassmann manifolds," *ACM Trans. Knowl. Discovery Data*, vol. 12, no. 1, pp. 13:1–13:22, 2018.

[20] Z. Ren, Q. Sun, and D. Wei, "Multiple kernel clustering with kernel K-means coupled graph tensor learning," in *Proc. 35th AAAI Conf. Artif. Intell.*, 2021, pp. 9411–9418.

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

[22] Z. Kang, L. Wen, W. Chen, and Z. Xu, "Low-rank kernel learning for graph-based clustering," *Knowl.-Based Syst.*, vol. 163, pp. 510–517, Jan. 2019.

[23] Z. Ren, H. Li, C. Yang, and Q. Sun, "Multiple kernel subspace clustering with local structural graph and low-rank consensus kernel learning," *Knowl.-Based Syst.*, vol. 188, Jan. 2020, Art. no. 105040.

[24] M. Chen, L. Huang, C. Wang, and D. Huang, "Multi-view clustering in latent embedding space," in *Proc. 34th AAAI Conf. Artif. Intell.*, 2020, pp. 3513–3520.

[25] J. Dai, Z. Ren, Y. Luo, H. Song, and J. Yang, "Multi-view clustering with latent low-rank proxy graph learning," *Cognit. Comput.*, vol. 13, no. 4, pp. 1049–1060, Jul. 2021.

[26] F. Nie, J. Li, and X. Li, "Parameter-free auto-weighted multiple graph learning: A framework for multiview clustering and semi-supervised classification," in *Proc. 25th Int. Joint Conf. Artif. Intell.*, 2016, pp. 1881–1887.

[27] 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, 2021.

[28] 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.*, vol. 32, no. 1, pp. 92–104, Jan. 2022.

[29] G.-F. Lu, H. Li, Y. Wang, and G. Tang, "Multi-view subspace clustering with Kronecker-basis-representation-based tensor sparsity measure," *Mach. Vis. Appl.*, vol. 32, no. 6, pp. 1–12, Nov. 2021.

[30] S. Wang, Y. Chen, Y. Jin, Y. Cen, Y. Li, and L. Zhang, "Error-robust low-rank tensor approximation for multi-view clustering," *Knowl.-Based Syst.*, vol. 215, Mar. 2021, Art. no. 106745.

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

[32] Z. Ren, M. Mukherjee, M. Bennis, and J. Lloret, "Multikernel clustering via non-negative matrix factorization tailored graph tensor over distributed networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7, pp. 1946–1956, Jul. 2021.

[33] Y. Chen, X. Xiao, and Y. Zhou, "Multi-view subspace clustering via simultaneously learning the representation tensor and affinity matrix," *Pattern Recognit.*, vol. 106, Oct. 2020, Art. no. 107441.

[34] 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. (ICCV)*, Dec. 2015, pp. 1582–1590.

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

[36] Y. Zhao, Y. Yun, X. Zhang, Q. Li, and Q. Gao, "Multi-view spectral clustering with adaptive graph learning and tensor schatten p-norm," *Neurocomputing*, vol. 468, pp. 257–264, Jan. 2022.

[37] C. Zhang, H. Fu, J. Wang, W. Li, X. Cao, and Q. Hu, "Tensorized multi-view subspace representation learning," *Int. J. Comput. Vis.*, vol. 128, nos. 8–9, pp. 2344–2361, Sep. 2020.

[38] 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, Dec. 2020.

[39] B. Wang, Y. Hu, J. Gao, Y. Sun, and B. Yin, "Low rank representation on Grassmann manifolds," in *Proc. Asian Conf. Comput. Vis.*, 2014, pp. 81–96.

[40] X. Piao, Y. Hu, J. Gao, Y. Sun, and B. Yin, "Double nuclear norm based low rank representation on Grassmann manifolds for clustering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12067–12076.

[41] J. Guo, Y. Sun, J. Gao, Y. Hu, and B. Yin, "Low rank representation on product Grassmann manifolds for multi-view subspace clustering," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2020, pp. 907–914.

[42] B. Wang, Y. Hu, J. Gao, Y. Sun, F. Ju, and B. Yin, "Learning adaptive neighborhood graph on Grassmann manifolds for video/image-set subspace clustering," *IEEE Trans. Multimedia*, vol. 23, pp. 216–227, 2021.

[43] 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.

[44] V. M. Patel and R. Vidal, "Kernel sparse subspace clustering," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 2849–2853.

[45] G.-Y. Zhang, X.-W. Chen, Y.-R. Zhou, C.-D. Wang, D. Huang, and X.-Y. He, "Kernelized multi-view subspace clustering via auto-weighted graph learning," *Int. J. Speech Technol.*, vol. 52, no. 1, pp. 716–731, Jan. 2022.

[46] S. El Hajjar, F. Dornaika, F. Abdallah, and N. Barrena, "Consensus graph and spectral representation for one-step multi-view kernel based clustering," *Knowl.-Based Syst.*, vol. 241, Apr. 2022, Art. no. 108250.

[47] T. Qiu and Y. Li, "Enhancing in-tree-based clustering via distance ensemble and kernelization," *Pattern Recognit.*, vol. 112, Apr. 2021, Art. no. 107731.

[48] X. Zhang, Z. Ren, H. Sun, K. Bai, X. Feng, and Z. Liu, "Multiple kernel low-rank representation-based robust multi-view subspace clustering," *Inf. Sci.*, vol. 551, pp. 324–340, Apr. 2021.
[49] P. Goyal and E. Ferrara, "Graph embedding techniques, applications, and performance: A survey," *Knowl.-Based Syst.*, vol. 151, pp. 78–94, Jul. 2018.
[50] C. Yang, Z. Ren, Q. Sun, M. Wu, M. Yin, and Y. Sun, "Joint correntropy metric weighting and block diagonal regularizer for robust multiple kernel subspace clustering," *Inf. Sci.*, vol. 500, pp. 48–66, Oct. 2019.
[51] 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.
[52] 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, Apr. 2019.
[53] F. Nie, X. Wang, M. Jordan, and H. Huang, "The constrained Laplacian rank algorithm for graph-based clustering," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 1969–1976.
[54] R. Xia, Y. Pan, L. Du, and J. Yin, "Robust multi-view spectral clustering via low-rank and sparse decomposition," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 2149–2155.
[55] F.-F. Li and P. Perona, "A Bayesian hierarchical model for learning natural scene categories," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2005, pp. 524–531.
[56] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 2, 2001, pp. 849–856.
[57] 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, Jan. 2013.
[58] 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. (CVPR)*, Jun. 2015, pp. 586–594.
[59] Y. Mei, Z. Ren, B. Wu, Y. Shao, and T. Yang, "Robust graph-based multi-view clustering in latent embedding space," *Int. J. Mach. Learn. Cybern.*, vol. 13, no. 2, pp. 1354–13042, 2021.

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

ZHENWEN REN (Member, IEEE) received the Ph.D. degree in control science and engineering from the Nanjing University of Science and Technology (NJUST), Nanjing, China, in 2021. He is working with the School of National Defence Science and Technology, Southwest University of Science and Technology (SWUST). He has published over 50 peer-reviewed papers, including those in highly regarded conferences and journals, such as the AAAI, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, the IEEE TRANSACTIONS ON CYBERNETICS, the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, and the *Pattern Recognition*. His research interests include computer vision, machine learning, deep learning, and industrial software.

BIN WU received the B.E. degree from Central South University, Changsha, China, in 1985, and the M.S. and Ph.D. degrees from the University of Science and Technology Beijing, Beijing, China, in 1993 and 1999, respectively. He is a Professor with the Southwest University of Science and Technology. His research interests include the most economical intelligent control and image processing.

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

YANHUA SHAO received the M.S. degree in pattern recognition and intelligent systems from the Southwest University of Science and Technology, China, in 2010, and the Ph.D. degree in instrument science and technology from Chongqing University. His research interests include machine learning and computer vision.