# Late Fusion Multiple Kernel Clustering with Proxy Graph Refinement

Siwei Wang, Xinwang Liu†, *Senior Member, IEEE,* Li Liu, *Senior Member, IEEE,* Sihang Zhou†, and En Zhu

***Abstract*—Multiple kernel clustering (MKC) optimally utilizes a group of pre-specified base kernels to improve clustering performance. Among existing MKC algorithms, the recently proposed late fusion MKC methods demonstrate promising clustering performance in various applications and enjoy considerable computational acceleration. However, we observe that the kernel partition learning and late fusion processes are separated from each other in the existing mechanism, which may lead to suboptimal solutions and adversely affect the clustering performance. In this paper, we propose a novel Late Fusion Multiple Kernel Clustering with Proxy Graph Refinement (LFMKC-GPR) framework to address these issues. Firstly, we theoretically revisit the connection between late fusion kernel base partition and traditional spectral embedding. Based on this observation, we construct a proxy self-expressive graph from kernel base partitions. The proxy graph in return refines the individual kernel partitions and also captures partition relations in graph structure rather than simple linear transformation. We also provide theoretical connections and considerations between the proposed framework and the multiple kernel subspace clustering. An alternate algorithm with proved convergence is then developed to solve the resultant optimization problem. After that, extensive experiments are conducted on twelve multi-kernel benchmark datasets, and the results demonstrate the effectiveness of our proposed algorithm. The code of the proposed algorithm is publicly available at https://github.com/wangsiwei2010/graphlatefusion_MKC.**

***Index Terms*—Multiple kernel clustering, Multi-view learning, Data fusion.**

## I. Introduction

CLUSTERING is one of the fundamental unsupervised learning tasks in data science and machine learning community. In the era of big data, data are often collected from multiple sources or domains as single-view information could not contain comprehensive information, which gives rise to multi-view clustering in literature. For example, for image clustering, images are often described by edge features, HOG features and Local Binary Pattern (LBP) features. Existing multi-view clustering can be roughly categorized into aspects:multiple-view subspace, co-training, multi-view ensemble clustering and multiple kernel clustering. Multi-view subspace clustering (MVSC) aims to seek unified subspaces from fused multi-view data representation and then separates data in the corresponding subspace. By capturing nonlinear structure and preserving pairwise similarity in graphs, MVSC has been widely applied in various applications, e.g. image classification, face clustering, community detection [1]–[7]. Multi-view ensemble clustering optimizes the optimal clustering partition matrix by aggregating a set of given pre-defined multiple partitions [8]–[10]. As an important extension to $k$-means to handle multi-view data, multiple kernel clustering (MKC) cooperates a group of weighted kernels from a given library to enhance clustering performances on non-linearly spreadable data. The existing approaches in literature can be roughly categorized into two strategies from the perspective of different fusion stages, i.e., kernel fusion and late fusion. The kernel fusion methods combine complementary information from multiple kernels and perform kernel $k$-means on the optimal kernel [11]–[20]. For example, a multiple kernel $k$-means algorithm is proposed to jointly optimize kernel weights, dimension reduction and clustering task [1]. The work in [2] suggests multiple data-dependent kernels to preserve local structures among different views. Then Liu et al. propose multiple kernel $k$-means with a matrix-induced regularization term to encourage the diversity of selected kernels [3]. Moreover, a multiple kernel $k$-means method with cluster-aware weighting is introduced in multi-view clustering [21].

Recently, late fusion based MKC is proposed to utilize

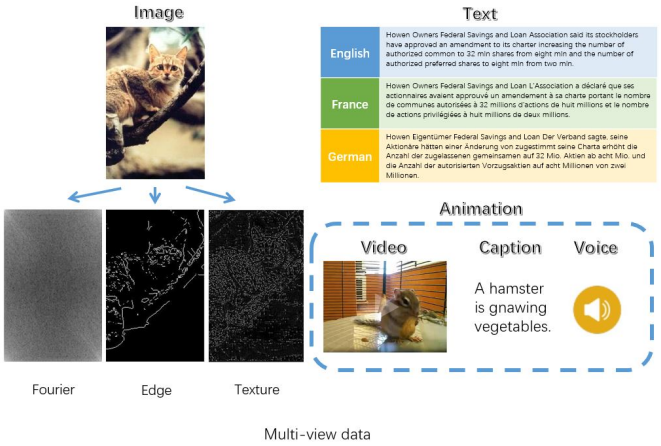



Fig. 1: An example of multi-view data. Images are often described by edge features, Fourier features and texture features. Text with the same information can be translated into multiple languages. Moreover, videos have vision, text and voice features.

S. Wang, X. Liu, and E. Zhu are with School of Computer, National University of Defense Technology, Changsha, China, 410073 (e-mail: {xinwangliu, enzhu}@nudt.edu.cn).)

L. Liu is with College of System Engineering, National University of Defense Technology, Changsha, China, and also with the Center for Machine Vision and Signal Analysis, University of Oulu, 90014 Oulu, Finland (E-mail: li.liu@oulu.fi).

S. Zhou is with College of Intelligence Science and Technology, National University of Defense Technology, Changsha, China, 410073 (e-mail: sihangjoe@gmail.com).

† Corresponding authors.

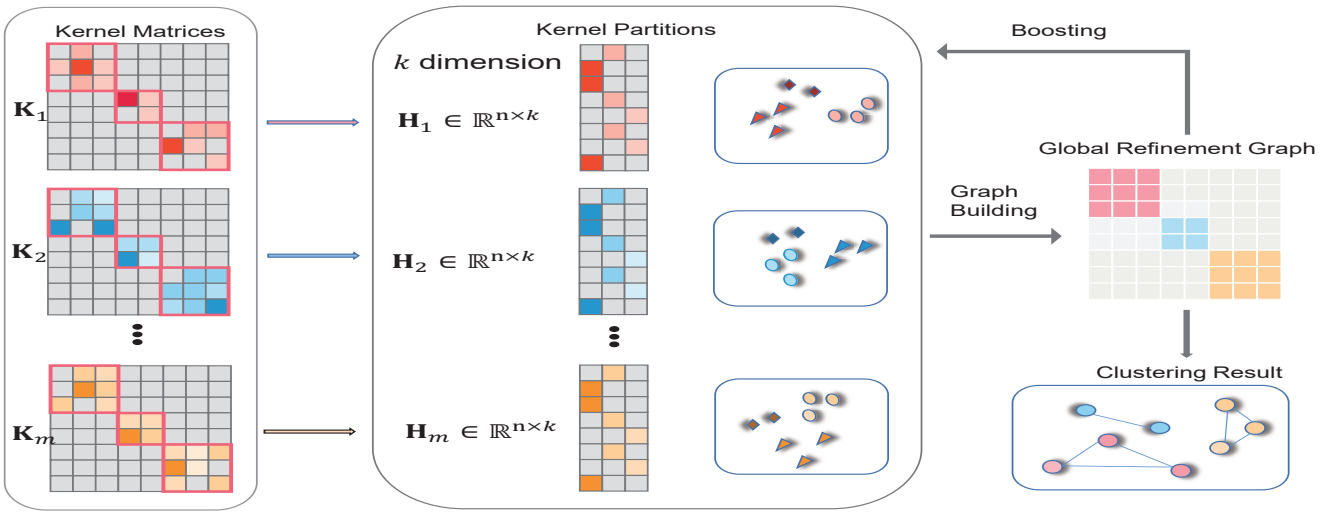



Fig. 2: The illustration of our proposed LFMKC-PGR. The late fusion kernel base partitions are initialized with kernel $k$-means performed on each kernel matrix. After that, a global self-expressive proxy graph is constructed to capture their complex partition structure. Then the kernel base partitions and proxy graph are alternately boosted until best serving for clustering.

underlying shared kernel partition by fusing partition level information, which significantly reduces the computation burden and avoids low-quality solutions [22]–[28]. Wang et al. efficiently obtain a unified kernel partition by maximizing its alignments with individual partitions [24]. Comparing to former kernel fusion methods, late fusion variants take advantages of partition information and enjoy considerable algorithm acceleration. Although the proposed late fusion methods enjoy low complexity and considerable promising performance in applications, they can still be improved from the following considerations: i)*The kernel base partition learning stage and the subsequent late fusion are separated from each other in the existing mechanism.* Therefore, their performance is highly dependent on the quality of pre-calculated kernel partitions in each view, which may contain noises or outliers to degrade performance and lead to sub-optimal solutions. ii)*These methods consider the relationships between kernel base partitions and consensus partition are linear transformation. However, this assumption might fail to handle real multi-kernel applications due to obstructions existing in data.* As consequence, these two major factors inhibit late fusion multiple kernel clustering from obtaining better performance.

To address these issues, in this paper, we propose to jointly optimize kernel base partitions and late fusion stage in a unified manner, which is termed as Late Fusion Multiple Kernel Clustering with Proxy Graph Refinement (LFMKC-PGR). Firstly, we theoretically illustrate the connection between kernel base partition and traditional spectral embedding under certain kernel conditions. Therefore, followed by traditional graph-based methods, we construct a proxy self-expressive graph for individual kernel base partitions and combine them into joint optimization. Moreover, by optimizing a shared self-expression matrix for base partitions to capture nonlinear relationships, they can be jointly negotiated with each other and reach a consensus on partition space best serving for clustering. In addition, extensive experiments on twelve multiple-view benchmark datasets are conducted to evaluate the effectiveness and efficiency of our proposed method. As demonstrated, the proposed algorithm enjoys superior clustering performance in comparison with several state-of-the-art multi-view kernel-based clustering methods.

The main contributions of this paper can be summarized as follows,

- We theoretically reveal that late fusion kernel partition can be regarded as spectral embedding under certain conditions. Based on that, traditional graph-based methods can be continually applied into late fusion multiple kernel clustering which gives a novel insight into MKC community.
- We unify the kernel base partition learning and late fusion refinement into one framework. Therefore they can be jointly promoted and reach a consensus on partition space best serving for clustering. Moreover, we theoretically uncover the proposed method with the existing multiple kernel subspace clustering framework and discuss their pros and cons.
- Extensive experiments are conducted on twelve multi-kernel benchmark datasets. By virtue of the proposed algorithm, LFMKC-PGR shows clear superiority over other multiple kernel state-of-the-art methods.

The rest of this paper is organized as follows. Section II outlines the related work of multiple kernel clustering. Section III presents the proposed optimization objective and the two-step alternate algorithm. Further, we also provide an analysis of the convergence and the computational complexity of our proposed algorithm. Section IV shows the experiment results with evaluation. Section V concludes the paper.

## II. Related Work

In this section, we introduce existing work most related to our study in this paper including kernel clustering and advanced multiple-kernel clustering methods.

### A. Multi-kernel k-means (MKKM)

In recent years, enormous multiple kernel clustering methods have been proposed to enhance task performance in literature,i.e., the co-training style methods, kernel fusion and late fusion strategies.

The co-training approaches for MKC iteratively obtain clustering results that can provide predicted clustering indices for the unlabeled data for other views. In this way, besides extracting the specific cluster information from the corresponding view, the clustering results are forced to be consistent across views. These methods may suffer performance degradation when the pseudo-labels obtained from other views are not reliable.

TABLE I: Common kernel functions

| Name | Expression | Parameter |
|---|---|---|
| Linear kernel | $\kappa(x_i, x_j) = x_i^T x_j$ | |
| Polynomial kernel | $\kappa(x_i, x_j) = (x_i^T x_j)^d$ | $d \geq 1$ is the degree of the polynomial |
| Gaussian kernel | $\kappa(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$ | $\sigma > 0$ is the bandwidth of the Gaussian kernel |
| Laplace kernel | $\kappa(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|}{\sigma}\right)$ | $\sigma > 0$ |
| Sigmoid kernel | $\kappa(x_i, x_j) = \tanh\left(\beta x_i^T x_j + \theta\right)$ | tanh is the hyperbolic tangent function,$\beta > 0, \theta < 0$ |

Kernel fusion based algorithms mainly optimize kernel coefficients for a group of kernel candidates [6], [29]–[32]. Let $\{\mathbf{x}_i\}_{i=1}^n \subseteq \mathcal{X}$ be a collection of $n$ samples, and $\phi_p(\cdot) : \mathbf{x} \in \mathcal{X} \mapsto \mathcal{H}_p$ be the $p$-th feature mapping that maps $\mathbf{x}$ onto a reproducing kernel Hilbert space $\mathcal{H}_p$ $(1 \leq p \leq m)$. In the multiple kernel setting, each sample is represented as $\phi_{\boldsymbol{\beta}}(\mathbf{x}) = [\beta_1\phi_1(\mathbf{x})^\top, \cdots, \beta_m\phi_m(\mathbf{x})^\top]^\top$, where $\boldsymbol{\beta} = [\beta_1, \cdots, \beta_m]^\top$ consists of the coefficients of the $m$ base kernels $\{\kappa_p(\cdot,\cdot)\}_{p=1}^m$. These coefficients will be optimized during learning. The relative commonly-used kernel functions are shown in Table I.

Based on the definition of $\phi_{\boldsymbol{\beta}}(\mathbf{x})$, a kernel function can be expressed as

$$\kappa_{\boldsymbol{\beta}}(\mathbf{x}_i, \mathbf{x}_j) = \phi_{\boldsymbol{\beta}}(\mathbf{x}_i)^\top \phi_{\boldsymbol{\beta}}(\mathbf{x}_j) = \sum\nolimits_{p=1}^m \beta_p^2 \kappa_p(\mathbf{x}_i, \mathbf{x}_j). \quad (1)$$

A kernel matrix $\mathbf{K}_{\boldsymbol{\beta}}$ is then calculated by applying the kernel function $\kappa_{\boldsymbol{\beta}}(\cdot,\cdot)$ into $\{\mathbf{x}_i\}_{i=1}^n$. Based on the kernel matrix $\mathbf{K}_{\boldsymbol{\beta}}$, the objective of MKKM can be written as

$$\begin{aligned} \min_{\mathbf{H},\boldsymbol{\beta}} \ & \mathrm{Tr}(\mathbf{K}_{\boldsymbol{\beta}}(\mathbf{I}_n - \mathbf{H}\mathbf{H}^\top)) \\ s.t. \ & \mathbf{H} \in \mathbb{R}^{n\times k}, \ \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k, \\ & \boldsymbol{\beta}^\top \mathbf{1}_m = 1, \ \beta_p \geq 0, \ \forall p. \end{aligned} \quad (2)$$

where $\mathbf{I}_k$ is an identity matrix with size $k \times k$.

The optimization problem in Eq.(2) can be solved by alternately updating $\mathbf{H}$ and $\boldsymbol{\beta}$:

i) **Optimizing $\mathbf{H}$ given $\boldsymbol{\beta}$**. With the kernel coefficients $\boldsymbol{\beta}$ fixed, $\mathbf{H}$ can be obtained by solving a kernel $k$-means clustering optimization problem shown in Eq.(3);

$$\begin{aligned} \max_{\mathbf{H}} \ & \mathrm{Tr}(\mathbf{H}^\top \mathbf{K}_{\boldsymbol{\beta}} \mathbf{H}) \\ s.t. \ & \mathbf{H} \in \mathbb{R}^{n\times k}, \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k, \end{aligned} \quad (3)$$

The optimal $\mathbf{H}$ for Eq.(3) can be obtained by taking the $k$ eigenvectors respecting to the largest eigenvalues of $\mathbf{K}_{\boldsymbol{\beta}}$.

ii) **Optimizing $\boldsymbol{\beta}$ given $\mathbf{H}$**. With $\mathbf{H}$ fixed, $\boldsymbol{\beta}$ can be optimized via solving the following quadratic programming with linear constraints,

$$\begin{aligned} \min_{\boldsymbol{\beta}} \ & \sum_{p=1}^m \beta_p^2 \mathrm{Tr}(\mathbf{K}_p(\mathbf{I}_n - \mathbf{H}\mathbf{H}^\top)), \\ s.t. \ & \boldsymbol{\beta}^\top \mathbf{1}_m = 1, \ \beta_p \geq 0. \end{aligned} \quad (4)$$

Along with this line, many variants of MKKM have been proposed in the literature. The work in [1] proposes a three-step alternate algorithm to jointly perform kernel clustering, coefficients and dimension reduction. The work in [3] proposes a multiple kernel $k$-means clustering algorithm with matrix-induced regularization to reduce the redundancy and enhance the diversity of the pre-defined kernels. Furthermore, the local kernel alignment criterion has been applied to multiple kernel learning to enhance the clustering performance in [33].

### B. Late Fusion Multiple Kernel Clustering

Based on the assumption that the multiple kernels are expected to share a consensus partition matrix among partition levels, late fusion methods seek the optimal kernel partition by combing linearly-transformed base partitions obtained from single views [22], [24], [25]. Given $n$ samples in $k$ clusters among $m$ views, their optimization goal can be mathematically expressed as

$$\begin{aligned} \max_{\mathbf{H}^c, \{\mathbf{W}_i\}_{i=1}^m, \boldsymbol{\beta}} \ & \mathrm{Tr}(\mathbf{H}^{c\top} \textstyle\sum_{i=1}^m \beta_i \mathbf{H}_i \mathbf{W}_i) + \lambda \boldsymbol{\Omega}(\mathbf{H}^c), \\ s.t. \ & \mathbf{H}^{c\top}\mathbf{H}^c = \mathbf{I}_k, \mathbf{W}_i^\top \mathbf{W}_i = \mathbf{I}_k, \\ & \|\boldsymbol{\beta}\|_2 = 1, \beta_i \geq 0, \ \forall i, \end{aligned} \quad (5)$$

where the first term and $\boldsymbol{\Omega}(\cdot)$ denote the late fusion alignment and regularization term for the consensus partition $\mathbf{H}^c$ respectively. $\mathbf{H}_i \in \mathbb{R}^{n\times k}$ and $\mathbf{W}_i$ are the $i$-th kernel partition matrix obtained from $i$-th kernel and its transformation matrix regarding the consensus partition matrix.

Although Eq. (5) accomplishes multiple kernel clustering with kernel individual partition matrices fusion via an effective and efficient manner, its partition presentation learning and late fusion are conducted separately which may leads to sub-optimal solutions. Moreover, the linear transformation relationships do not always hold when facing with noises or outliers in real-world complex data. As a result, these two factors shadow the representation ability of latent kernel partitions and adversely harm the performance of the model. In the following, we propose a proxy graph to refine the base partitions and optimally optimize them and fusion in a unified manner termed Late Fusion Multiple Kernel Clustering with Proxy Graph Refinement (LFMKC-PGR).

## III. Late Fusion Multiple Kernel Clustering with Proxy Graph Refinement

In this section, we firstly describe our proposed LFMKC-PGR in details. Then an efficient two-step optimization algorithm is proposed to solve the respective optimization formula. Finally, we summarize our algorithm and provide analysis and extensions for LFMKC-PGR.

### A. Revisit Kernel $k$-means and Spectral Clustering

In this section, we firstly revisit the popular kernel $k$-means clustering and mathematically reveal its connection with traditional spectral clustering. Given a similarity matrix $\mathbf{W}$, the optimization goal of spectral clustering algorithm can be rewritten as [34],

$$\begin{aligned} &\min_{\mathbf{F}} \ \mathrm{Tr}(\mathbf{F}^\top \mathbf{L}_{\mathbf{W}} \mathbf{F}), \\ &s.t. \ \ \mathbf{F} \in \mathbb{R}^{n\times k}, \mathbf{F}^\top \mathbf{F} = \mathbf{I}_k, \end{aligned} \tag{6}$$

where $\mathbf{F}$ is regarded as the spectral embedding of data matrix and $\mathbf{L}$ is the Laplacian matrix for the respective affinity matrix $\mathbf{W}$ as $\mathbf{L}_{\mathbf{W}} = \mathbf{D} - \mathbf{W}$, where $\mathbf{D}$ is the degree matrix.

The traditional kernel $k$-means clustering can be mathematically written as follows [35],

$$\begin{aligned} &\max_{\mathbf{H}} \ \mathrm{Tr}(\mathbf{H}^\top \mathbf{K} \mathbf{H}), \\ &s.t. \ \ \mathbf{H} \in \mathbb{R}^{n\times k}, \mathbf{H}^\top \mathbf{H} = \mathbf{I}_k, \end{aligned} \tag{7}$$

It seems that there is no significant connection between Eq. (6) and Eq. (7) at first glance. The following Theorem gives a theoretic analysis of kernel $k$-means and Eq. (6).

**Theorem 1.** *Given a normalized kernel matrix $\mathbf{K}$ as the affinity matrix under the condition $\mathbf{D}_K = \mathbf{I}_n$, the optimal solutions of $\mathbf{F}^*$ in Eq. (6) and $\mathbf{H}^*$ in Eq. (7) satisfy the following equation $\mathbf{F}^* = \mathbf{H}^*$.*

*Proof.* Notice that $\mathrm{Tr}(\mathbf{F}^\top \mathbf{L}_{\mathbf{K}} \mathbf{F}) = \mathrm{Tr}(\mathbf{F}^\top (\mathbf{D}_{\mathbf{K}} - \mathbf{K})\mathbf{F}) = k - \mathrm{Tr}(\mathbf{F}^\top \mathbf{K}\mathbf{F})$, where $k$ is a constant. The optimal solution for Eq. (6) is the $k$ smallest eigenvectors of $\mathbf{L}_{\mathbf{K}}$ while the solution for Eq. (7) is the $k$ largest eigenvectors of $\mathbf{K}$. Therefore it is straightforward to see that $\mathbf{F}^* = \mathbf{H}^*$. The equation holds if we set the degree matrix $\mathbf{D}_{\mathbf{K}}$ as $\mathbf{I}_n$ and this could be easily done by normalizing the kernel matrix. □

Theorem 1 inspires us a new perspective on Eq. (5) that the kernel partitions $\{\mathbf{H}_i\}_{i=1}^m$ can be regarded as the spectral embeddings from individual views under certain conditions. Therefore they can be refined by the existing graph-based methods and jointly be optimized during the learning process. In the next subsection, we describe our proposed proxy graph refinement in details to combine kernel partition and graph constructing into one objective and further improve the existing late-fusion based strategy.

### B. The Proposed Formula

Regarding Eq. (5), the base partition matrices are learned individually from each kernel with fixed representations during the learning stage and the consensus $\mathbf{H}^c$ is obtained by linear transformation. Therefore, the kernel representation learning and the fusion procedures are conducted separately which do not satisfy an end-to-end manner. Moreover, we might capture more complicated relationships between each base partitions rather than simple linear transformations.

The kernel base partitions $\{\mathbf{H}_i\}_{i=1}^m$ are independently initialized from each kernel in original model of Eq. (5). Different from that, the base partitions are refined by a proxy graph regularization term in our new model. Inspired by the self-expressive subspace graph building method [32], [36]–[46], we treat each base partition with refined similarity graph $\mathbf{S}$ building as follows,

$$\begin{aligned} &\min_{\mathbf{S}} \sum_{i=1}^{m} \|\mathbf{H}_i - \mathbf{S}\mathbf{H}_i\|_{\mathrm{F}}^2 + \beta \|\mathbf{S}\|_{\mathrm{F}}^2, \\ &s.t. \ \ \mathbf{S} \geq 0, \ \mathbf{S}\mathbf{1} = \mathbf{1}, diag(\mathbf{S}) = 0, \end{aligned} \tag{8}$$

where $\mathbf{S}$ is the shared proxy graph for base partitions and represents the complex relationship between each single view representation, and $\|\mathbf{S}\|_{\mathrm{F}}^2$ is the regularization term. $\mathbf{S} \geq 0$ ensures the non-negative of the similarity matrix § and $\mathbf{S}\mathbf{1} = \mathbf{1}$ normalizes the obtained $\mathbf{S}$. Moreover, $diag(\mathbf{S}) = 0$ avoids the trivial solution.

By minimizing Eq. (8), $\mathbf{S}_{ij}$ can be regarded as the similarity score between $i$-th and $j$-th sample. The larger value $\mathbf{S}_{ij}$ is, the more likely two samples belong to the same cluster. After getting the global graph $\mathbf{S}$, we refine the kernel base partition with the guidance of kernel matrices and the learned global graph. Our idea can be mathematically expressed as follows,

$$\begin{aligned} \min_{\{\mathbf{H}_i\}_{i=1}^m, \mathbf{S}} \ & \sum_{i=1}^{m} \underbrace{\mathrm{Tr}(\mathbf{K}_i(\mathbf{I} - \mathbf{H}_i\mathbf{H}_i^\top))}_{Kernel\ clustering} + \underbrace{\lambda \|\mathbf{H}_i - \mathbf{S}\mathbf{H}_i\|_{\mathrm{F}}^2}_{Graph\ Refinement} \\ & + \beta \|\mathbf{S}\|_{\mathrm{F}}^2, \\ & s.t. \ \ \mathbf{S} \geq 0, \ \mathbf{S}\mathbf{1} = \mathbf{1}, diag(\mathbf{S}) = 0, \mathbf{H}_i^\top \mathbf{H}_i = \mathbf{I}_k. \end{aligned} \tag{9}$$

From the above formula, we summarize the differences between our MKL methods and multi-view subspace clustering as follows: (i) Methods of multi-view subspace clustering are facing raw data or extracted features while MKL method optimizes the multiple kernel matrices (similarity matrices). Further, it is quite straightforward to combine MKL and feature selection into a unified framework. With an adequate feature selection strategy, the base kernels can be dynamically constructed from the selected features rather than raw data. (ii)The advantages of MKL is to handle with **nonlinear-separable** data. MKL adopts several kernel functions to transfer original data to their new representations in Hilbert Space. While they are not easy to be clustered in original space. (iii)MKL can naturally handle with **heterogeneous source information** regardless of the data items. Whatever the data types are, the kernels can be defined once the similarity measure is defined. Therefore MKL is widely applied in Biology and Chemistry. Other methods will conduct graph alignment to handle with in-heterogeneous multi-view information.

Our proposed LFMKC-PGR model jointly optimizes the individual kernel representations and the consensus proxy graph into a unified formula, which avoids the former separate two-step late fusion strategy. Although the formula is quite simple and straightforward, LFMKC-PGR has the following merits. (1) It addresses multiple kernel clustering via a refined late fusion manner which simultaneously combines kernel partition learning and graph refinement in a joint framework.(2)View-specific correlations are captured in graph structure making it more robust to noises or corrupted multi-view data.(3) More considerations of graph constructions in kernel partition space can be easily adjusted into this framework with prior knowledge.

### C. Connections with the Existing Multiple Kernel Subspace Clustering

Comparing with conventional MKC algorithms, ours is the first attempt to combine kernel k-means and the latter graph regularization which has not been studied in the existing literature. The equation seems much like with the combination of kernel self-expressive subspace clustering [14], [47]–[49].

The kernel self-expressive subspace clustering extends traditional subspace clustering with kernel tricks to handle nonlinearly separable subspaces [47]. The original formula is that

$$\min_{\mathbf{S}} \mathrm{Tr}(\mathbf{K}(\mathbf{I} - 2\mathbf{S} + \mathbf{S}\mathbf{S}^\top)), \tag{10}$$

where $\mathbf{S}$ is the learned kernel graph.

We summarize the differences of our new MKC framework comparing to the kernel subspace algorithms. As we mentioned in our paper, the original space may be infinite dimensional (e.g. Gaussian Kernel) and contain noise or outliers, which could better capture information in partition level. However, in our paper, we directly start from the kernel k-means and therefore the subsequent graph is constructed in the partition space rather than the former in the original Reproducing kernel Hilbert Space (RKHS). As can be seen, the graph $\mathbf{S}$ in [47] is only reflected by the original kernel $\mathbf{K}$ while ours is both influenced by kernel and the partition matrix $\mathbf{H}$ respectively. These lead to two different formulations in [47] and ours. To the best of our knowledge, it is also the first practice within MKC domain. By following this new framework, more interesting approaches could be introduced into MKC and can greatly contribute to MKC community.

### D. Optimization

The optimization problem in Eq. (9) is a non-convex problem when regrading the two variables. In this section, we develop an alternate optimization algorithm which separates the resultant problem into two subproblems such that each is convex when the other variable is fixed,

*1) Update* $\{\mathbf{H}_i\}_{i=1}^m$*:* By fixing $\mathbf{S}$, $\{\mathbf{H}_i\}_{i=1}^m$ can be solved individually with each of the $m$ sub-problems. The optimization problem regrading of $\mathbf{H}_i$ can be simplified as,

$$\begin{aligned} &\min_{\mathbf{H}_i} \ \mathrm{Tr}(\mathbf{K}_i(\mathbf{I} - \mathbf{H}_i\mathbf{H}_i^\top)) + \lambda \left\|\mathbf{H}_i - \mathbf{S}\mathbf{H}_i\right\|_{\mathrm{F}}^2, \\ &s.t. \ \ \mathbf{H}_i^\top \mathbf{H}_i = \mathbf{I}_k, \end{aligned} \tag{11}$$

which can be further converted into

$$\begin{aligned} &\max_{\mathbf{H}_i} \ \mathrm{Tr}((\mathbf{K}_i - \lambda(\mathbf{I} - 2\mathbf{S} + \mathbf{S}\mathbf{S}^\top))\mathbf{H}_i\mathbf{H}_i^\top)), \\ &s.t. \ \ \mathbf{H}_i^\top \mathbf{H}_i = \mathbf{I}_k. \end{aligned} \tag{12}$$

By denoting $\mathbf{G} = \mathbf{K}_i - \lambda(\mathbf{I} - 2\mathbf{S} + \mathbf{S}\mathbf{S}^\top)$, the optimal $\mathbf{H}_i$ in Eq. (11) can be obtained by taking the $k$ largest eigenvectors corresponding to the largest $k$ eigenvalues of $\mathbf{G}$.

*2) Update* $\mathbf{S}$*:* When $\mathbf{H}_i$ being fixed, Eq. (9) can be rewritten as,

$$\begin{aligned} &\min_{\mathbf{S}} \sum_{i=1}^m \lambda \left\|\mathbf{H}_i - \mathbf{S}\mathbf{H}_i\right\|_{\mathrm{F}}^2 + \beta \left\|\mathbf{S}\right\|_{\mathrm{F}}^2, \\ &s.t. \quad \mathbf{S} \geq 0, \ \mathbf{S}\mathbf{1} = \mathbf{1}, diag(\mathbf{S}) = 0, \end{aligned} \tag{13}$$

Specially, we design a two-step algorithm to quickly solve Eq. (13). In the first step, we solve Eq. (13) without constraints, which can be written as,

$$\hat{\mathbf{S}} = \arg\min_{\mathbf{S}} \sum_{i=1}^m \lambda \left\|\mathbf{H}_i - \mathbf{S}\mathbf{H}_i\right\|_{\mathrm{F}}^2 + \beta \left\|\mathbf{S}\right\|_{\mathrm{F}}^2, \tag{14}$$

The Eq. (14) is a constraint-free problem. By taking the derivation of Eq. (14) with respect to $\mathbf{S}$ to zero, we can get the closed-form solution to Eq. (14),

$$\hat{\mathbf{S}} = \left(\mathbf{C} + \frac{\beta}{\lambda}\mathbf{I}\right)^{-1} \mathbf{C}, \tag{15}$$

where $\mathbf{C} = \sum_i^m \mathbf{H}_i\mathbf{H}_i^\top$.

Then, we can obtain the approximate solution of $\mathbf{S}$ by projecting $\hat{\mathbf{S}}_i$ through the following minimization problem with proper constraints:

$$\min_{\mathbf{S}\geq 0, \ \mathbf{S}\mathbf{1}=\mathbf{1}, diag(\mathbf{S})=0} \left\|\mathbf{S} - \hat{\mathbf{S}}\right\|_{\mathrm{F}}^2, \tag{16}$$

This problem yields a close-formed solution that

$$\mathbf{S}_{j,:} = \max\left(\hat{\mathbf{S}}_{j,:} + \alpha_j \mathbf{1}, 0\right), \mathbf{S}_{jj} = 0, \alpha_j = \frac{1 + \hat{\mathbf{S}}_{j,:}^\top \mathbf{1}}{n}, \tag{17}$$

*Proof.* The problem of Eq. (16) can be easily rewritten into $n$ row-formed independent optimization problems as follows,

$$\min_{\mathbf{S}_{j,:}\geq 0, \ \mathbf{S}_{j,:}^\top \mathbf{1}=1, \mathbf{S}_{jj}=0} \left\|\mathbf{S}_{j,:} - \hat{\mathbf{S}}_{j,:}\right\|_{\mathrm{F}}^2, \tag{18}$$

where $\mathbf{S}_{j,:}$ is the $j$-th row of $\mathbf{S}$. We write the he Lagrangian function of Eq. (18) as,

$$\mathcal{L}\left(\mathbf{S}_{j,:}, \alpha, \beta\right) = \left\|\mathbf{S}_{j,:} - \hat{\mathbf{S}}_{j,:}\right\|_F^2 - \alpha_j\left(\mathbf{S}_{j,:}^\top \mathbf{1} - 1\right) - \eta_j^\top \mathbf{S}_{j,:}, \tag{19}$$

where $\alpha$ and $\eta_j$ are the respective Lagrangian multipliers. Then the KKT conditions are written as,

$$\begin{cases} \mathbf{S}_{j,:} - \hat{\mathbf{S}}_{j,:} - \alpha_j \mathbf{1} - \eta_j = 0, \\ \eta_j \bigodot \mathbf{S}_{j,:} = 0, \end{cases} \tag{20}$$

Therefore with $\mathbf{S}_{j,:}^\top \mathbf{1} = 1, \mathbf{S}_{jj} = 0$, we can easily obtain that

$$\mathbf{S}_{j,:} = \max\left(\hat{\mathbf{S}}_{j,:} + \alpha_j \mathbf{1}, 0\right), \mathbf{S}_{jj} = 0, \alpha_j = \frac{1 + \hat{\mathbf{S}}_{j,:}^\top \mathbf{1}}{n}, \tag{21}$$

This completes the proof. □

**Algorithm 1:** Late Fusion Multiple Kernel Clustering with Proxy Graph Refinement (LFMKC-PGR)

**Input:** Base kernel matrices $\{\mathbf{K}_i\}_{i=1}^m$, clustering number $k$, Hyper-parameters $\lambda, \beta$.
**Initialize:** $\mathbf{S}$
**while** *not converged* **do**
  Update $\{\mathbf{H}_i\}_{i=1}^m$ by solving Eq. (12);
  Update $\mathbf{S}$ by obtaining Eq. (17);
**end**
**Output:** Performing spectral clustering on $\mathbf{S}$.

### E. Analysis and Discussions

*Computational Complexity*: With the optimization process outlined in Algorithm 1, the total time complexity consists of two parts referring to the alternate steps. The first step mentioned in Eq. (12), actually needs singular value decomposition (SVD) for $\mathbf{G}$ and therefore needs $\mathcal{O}(mn^2k)$. As for the third step, we design a twp-step approximate algorithm for solving $\mathbf{S}$. Its time complexity is $\mathcal{O}(n^2k)$. The key issue in time complexity in the algorithm is to solve the inverse of $C + \frac{\beta}{\lambda}I$ with the size $n * n$. Notice that to obtain $C$ needs $O(n^2k)$.Then we formalize $C = UU^\top (O(n^2k)$, with the size $n * k$). We apply the Woodbury formulation which is widely applied in ridge regression to accelerate the inverse problem into $O(n^2k)$ with the following equation. $\left(\mathbf{C} + \frac{\beta}{\lambda}\mathbf{I}_n\right)^{-1} = \frac{\lambda}{\beta}\left(\frac{\lambda}{\beta}\mathbf{C} + \mathbf{I}_n\right)^{-1} = \frac{\lambda}{\beta}\left(\mathbf{U}\mathbf{U}^\top + \mathbf{I}_n\right)^{-1} = \frac{\lambda}{\beta}\left(\mathbf{I}_n - \mathbf{U}\left(\mathbf{I}_k + \mathbf{U}^\top\mathbf{U}\right)^{-1}\mathbf{U}^\top\right)$. Hence for each iteration, the whole time complexity of our proposed algorithms is $\mathcal{O}(mn^2k + n^2k)$.

*Convergence*: It is easy to obtain that the whole optimization function is lower bounded to 0. As the two subproblems are strictly convex when optimizing one variable and keeping the others fixed. The objective of Algorithm 1 is monotonically increased when optimizing one variable with the others fixed at each iteration. As a result, the proposed algorithm can be verified to be convergent according to [50].

*Extensions*: LFMKC-PGR offers a novel insight on the connection between kernel partitions and traditional spectral embedding. More interesting graph-based methods can be introduced to this framework. For example, local graph structure building strategy can be applied to further enhance the clustering performances by exploiting local structures among different views.

## IV. Experiment

In this section, we evaluate the effectiveness and efficiency of the proposed method for twelve widely used multi-view benchmark datasets with strong competitors from the perspectives of clustering performance, parameter sensitivity and convergence.

TABLE II: Datasets used in our experiments.

| Dataset | #Samples | #Kernels | #Classes |
|---|---|---|---|
| AR10P | 130 | 6 | 10 |
| YALE | 165 | 5 | 15 |
| ProteinFold | 694 | 12 | 27 |
| Flower17 | 1360 | 7 | 17 |
| Nonplant | 2732 | 69 | 3 |
| Flower102 | 8189 | 4 | 102 |
| Caltech102-5 | 510 | 48 | 102 |
| Caltech102-10 | 1020 | 48 | 102 |
| Caltech102-15 | 1530 | 48 | 102 |
| Caltech102-20 | 2040 | 48 | 102 |
| Caltech102-25 | 2550 | 48 | 102 |
| Caltech102-30 | 3060 | 48 | 102 |

### A. Datasets

The proposed algorithm is experimentally evaluated on twelve widely used multiple kernel benchmark datasets shown in Table II. They are AR10P,[1] Oxford Flower17 and Flower102[2], ProteinFold[3], YALE Face[4], Nonplant and Caltech102[5]. For these datasets, all kernel matrices are precomputed and can be publicly downloaded from the above websites. Further, followed by [22], we have downloaded the last six Caltech102 datasets where Caltech102-5 denotes the number of samples belonging to each cluster is 5 and so on.

### B. Compared Algorithms

In the experiments, our proposed algorithm is compared with the following state-of-the-art multiple kernel or subspace clustering methods. (1) **Best Single Kernel $k$-means (BSKM)** (2) **Multiple Kernel $k$-means (MKKM)** [51]: The algorithm alternatively performs kernel $k$-means and updates the kernel coefficients. (3) **Co-regularized Spectral Clustering (CRSC)** [52]: CRSC provides a co-regularization way to perform spectral clustering on multiple views. (4) **Robust Multiple Kernel $k$-means using $\ell_{2,1}$ norm (RMKKM)** [53]: RMKKM simultaneously finds the clustering label, the cluster membership and the optimal combination of multiple kernels by adding $\ell_{2,1}$ norm. (5) **Robust Multi-view Spectral Clustering (RMSC)** [54]: RMSC constructs a transition probability matrix from each single view, and then use recover a shared low-rank transition probability matrix as an input to the standard Markov chain for clustering. (6) **Multiple Kernel $k$-means with Matrix-induced Regularization (MKMR)** [3]: MKMR fulfills the multiple kernel $k$-means clustering with a matrix-induced regularization to reduce the redundancy and enhance the diversity of the kernels. (7) **Multiple Kernel Clustering with Local Kernel Alignment Maximization (MKAM)** [33]: The algorithm maximizes the proposed local kernel alignment and therefore captures local structure among kernels. (8) **Multi-view Clustering via Late Fusion Alignment Maximization (MLFA)** [24]: MLFA maximizes the alignment of individual kernel partitions and consensus one, and reach an agreement on partition level information (9) **Flexible Multi-View Representation Learning for Subspace Clustering (FMR)** [55]: FMR optimizes subspace clustering via encoding complementary latent representations and their nonlinear or high-order correlations from multiple views

### C. Experimental Setting

For all the above mentioned algorithms, we have downloaded their public Matlab code implementations from original websites.The hyper-parameters are set according to the suggestions of the corresponding literature. For the proposed algorithm LFMKC-PGR, the trade-off parameters $\lambda$ and $\beta$ are chosen from $\left[2^{-2}, 2^{-1}, \cdots, 2^2\right]$ by grid search. For all

[1] http://featureselection.asu.edu/old /datasets.php
[2] http://www.robots.ox.ac.uk/~vgg/data/flowers/
[3] http://mkl.ucsd.edu/dataset/protein-fold-prediction
[4] www.cs.yale.edu/cvc/projects/yalefaces/yalefaces.html
[5] http://www.vision.caltech.edu/archive.html

TABLE III: The ACC, NMI and Purity comparison of different clustering algorithms on twelve benchmark datasets. The best result is highlighted and boldfaced with underlines.

| Dataset | Metric | BSKM | MKKM | CRSC | RMKKM | RMSC | MKMR | MKAM | MLFA | FMR | Proposed |
|---|---|---|---|---|---|---|---|---|---|---|---|
| AR10P | ACC | 43.08 | 40.00 | 38.46 | 30.77 | 30.77 | 39.23 | 27.69 | 41.54 | 51.23 | **56.15** |
| | NMI | 42.61 | 39.53 | 39.82 | 26.62 | 27.87 | 40.11 | 24.72 | 39.15 | 45.52 | **51.82** |
| | Purity | 43.08 | 40.00 | 39.23 | 32.31 | 33.08 | 39.23 | 28.46 | 41.54 | 51.23 | **56.15** |
| YALE | ACC | 56.97 | 52.12 | 56.97 | 56.36 | 58.03 | 60.00 | 46.67 | 54.55 | 61.21 | **62.42** |
| | NMI | 58.42 | 54.16 | 57.69 | 59.32 | 57.58 | 62.87 | 53.51 | 59.86 | 60.31 | **63.48** |
| | Purity | 57.58 | 52.73 | 57.58 | 58.18 | 57.24 | 60.00 | 49.09 | 55.76 | 61.33 | **62.42** |
| ProteinFold | ACC | 33.86 | 27.23 | 34.87 | 30.98 | 33.00 | 36.46 | 37.90 | 35.88 | 34.96 | **40.06** |
| | NMI | 42.03 | 37.16 | 43.32 | 38.78 | 43.91 | 45.32 | 44.46 | 44.00 | 43.68 | **48.72** |
| | Purity | 41.21 | 33.86 | 40.78 | 36.60 | 42.36 | 42.65 | 43.95 | 41.93 | 42.22 | **45.97** |
| Flower17 | ACC | 42.06 | 45.37 | 52.35 | 53.38 | 51.10 | 58.82 | 57.87 | 60.16 | 58.78 | **62.28** |
| | NMI | 45.14 | 45.35 | 50.42 | 52.56 | 54.39 | 57.05 | 56.06 | 59.79 | 56.98 | **61.72** |
| | Purity | 44.63 | 46.84 | 53.01 | 55.07 | 54.12 | 60.51 | 59.26 | 62.13 | 59.66 | **63.60** |
| Nonplant | ACC | 49.38 | 54.32 | 55.56 | 49.33 | 60.65 | 56.59 | 59.57 | 50.07 | 36.70 | **67.50** |
| | NMI | 16.55 | 15.83 | 17.44 | 16.55 | 20.35 | 23.43 | 23.04 | 16.55 | 0.50 | **25.56** |
| | Purity | 72.18 | 71.45 | 73.17 | 72.18 | 70.50 | 73.33 | 74.34 | 72.18 | 60.36 | **75.29** |
| Flower102 | ACC | 33.13 | 21.96 | 37.26 | 28.17 | 32.97 | 39.91 | 40.84 | 42.73 | 35.24 | **46.78** |
| | NMI | 48.99 | 42.30 | 54.18 | 48.17 | 53.36 | 57.27 | 57.60 | 57.59 | 57.42 | **60.30** |
| | Purity | 38.78 | 27.61 | 44.08 | 33.86 | 40.24 | 46.39 | 48.21 | 49.73 | 41.62 | **53.07** |
| Caltech-5 | ACC | 36.86 | 28.63 | 36.08 | 32.75 | 33.73 | 38.04 | 32.16 | 37.45 | 36.27 | **43.73** |
| | NMI | 68.64 | 65.97 | 70.60 | 66.76 | 68.93 | 71.08 | 67.18 | 71.87 | 70.25 | **73.69** |
| | Purity | 36.24 | 29.80 | 37.65 | 33.92 | 34.90 | 39.02 | 33.92 | 39.61 | 37.29 | **45.49** |
| Caltech-10 | ACC | 30.88 | 22.75 | 33.43 | 26.67 | 29.80 | 33.73 | 28.33 | 32.45 | 28.73 | **40.78** |
| | NMI | 59.77 | 55.80 | 62.10 | 57.28 | 59.86 | 62.76 | 58.51 | 61.99 | 59.09 | **66.90** |
| | Purity | 31.24 | 24.22 | 35.29 | 28.82 | 31.47 | 35.88 | 30.39 | 34.22 | 29.80 | **43.73** |
| Caltech-15 | ACC | 29.11 | 20.39 | 31.18 | 24.90 | 25.49 | 32.29 | 27.32 | 31.11 | 17.12 | **39.93** |
| | NMI | 53.66 | 49.27 | 57.73 | 52.04 | 54.57 | 58.25 | 55.20 | 57.66 | 46.81 | **63.01** |
| | Purity | 31.81 | 21.63 | 33.14 | 26.21 | 27.12 | 34.25 | 28.89 | 33.14 | 17.71 | **41.83** |
| Caltech-20 | ACC | 28.20 | 18.73 | 30.98 | 24.51 | 23.87 | 32.55 | 25.88 | 30.44 | 9.71 | **37.25** |
| | NMI | 53.19 | 45.61 | 54.84 | 48.66 | 50.34 | 56.06 | 51.42 | 54.33 | 37.74 | **59.58** |
| | Purity | 31.91 | 20.39 | 32.50 | 26.13 | 25.59 | 34.66 | 27.84 | 32.60 | 10.22 | **39.85** |
| Caltech-25 | ACC | 26.41 | 16.63 | 29.69 | 21.92 | 24.08 | 30.12 | 26.16 | 29.45 | 8.23 | **36.47** |
| | NMI | 49.92 | 41.86 | 52.04 | 45.53 | 48.35 | 52.94 | 50.12 | 52.00 | 33.43 | **57.04** |
| | Purity | 30.41 | 18.00 | 31.57 | 23.45 | 25.80 | 32.20 | 28.75 | 31.49 | 8.64 | **38.75** |
| Caltech-30 | ACC | 25.91 | 16.31 | 28.53 | 21.41 | 22.58 | 31.31 | 24.54 | 28.56 | 7.50 | **36.37** |
| | NMI | 49.31 | 39.92 | 50.42 | 43.72 | 46.04 | 51.55 | 47.39 | 50.12 | 30.38 | **55.98** |
| | Purity | 28.71 | 18.04 | 30.07 | 23.50 | 24.15 | 33.20 | 26.76 | 29.87 | 7.75 | **38.40** |

datasets, we assumed that the true number of clusters is given. The widely used clustering accuracy (ACC), normalized mutual information (NMI) and purity are applied to evaluate the clustering performance. For all algorithms, we repeat each experiment for $50$ times with random initialization to reduce the effect of randomness caused by $k$-means, and report the best result. All our experiments are conducted on a desktop computer with a 2.5GHz Intel Platinum 8269CY CPU and 48GB RAM, MATLAB 2019b (64bit).

### *D. Experimental Results*

Table III presents the ACC comparison of the above algorithms on the twelve benchmark datasets. The best result is highlighted with underlines. Based on the results, we have the following observations:

- Our proposed algorithm shows clear advantages over other multi-kernel clustering baselines, with 12 best out of the total 12 datasets; in particular, the margins for the nine data sets: AR10P, Nonplant, Flower102 and the six Caltech are very impressive, outperforming the second-best algorithm 9.61%,11.29%,9.48%,14.95%,14.85%,15.94%,10.47% and 6.90% on ACC respectively. These results verify the effectiveness of the proposed method comparing to existing state-of-the-art approaches.
- Comparing with the FMR [55]) , the proposed LFMKC-PGR consistently further improves the clustering performance and achieves better results among the benchmark datasets. Both of them adopt the self-expressive subspaces for graph building. The clustering results clearly demonstrate that adapting kernel representations into proxy graphs might capture non-linearly separable data comparing to existing subspace methods.
- Mentioned before, our LFMKC-PGR originates from MLFA [24] which separates kernel base partition learning and the late fusion stage. As can be seen, the newly-proposed algorithm significantly surpasses MLFA in real experiments. Therefore, it is vital to jointly combine the kernel base partition learning and late fusion refinement in multiple kernel clustering.

We also report the NMI and purity in Table III. Again, we observe that the proposed algorithm significantly outperforms

other multiple kernel clustering algorithms. These results are consistent with our observations in Table III.

In summary, the above experimental results have well demonstrated the effectiveness of our proposed method comparing to other state-of-the-art methods. We attribute the superiority of the proposed algorithm as two aspects: i) LFMKC-PGR incorporates kernel base partition learning and proxy graph refinement into a unified framework. By the virtue of it, kernel partitions are refined by global proxy graph and subsequently contributing to construct better graphs. Therefore the two processes are mutually promoted serving for clustering. ii) Compared with the existing late fusion multiple kernel methods, the proposed LFMKC-GPR adopts graph structure to capture complex relationships between multiple partitions which is more suitable in real applications. These two factors contribute to significant improvements in clustering performance.

### *E. Comparing with Existing Multiple Kernel Subspace Clustering*

As mentioned in Section III-C, our proposed LFMKC-PGR has a close relationship with the existing multiple kernel subspace clustering methods which also construct graph from kernels. Therefore we also conduct comparison experiments on the existing STOA multi-kernel subspace clustering methods [14], which we refer it SPMKC in Table IV. And the representative deep multi-view clustering method DAMC [56] is also shown in the table.

TABLE IV: The ACC NMI and Purity comparison of multi-kernel subspace clustering, deep multi-view clustering methods and ours.

| Dataset | SPMKC | DAMC | Proposed |
|---|---|---|---|
| ACC(%) | | | |
| CCV | 23.64 | 20.51 | **26.89** |
| NonPlant | 13.50 | 54.90 | **67.50** |
| Flower17 | 50.07 | 30.29 | **62.28** |
| Flower102 | 36.55 | 22.50 | **46.78** |
| NMI(%) | | | |
| CCV | 28.11 | 22.50 | **20.72** |
| NonPlant | 1.27 | 16.64 | **25.56** |
| Flower17 | 51.28 | 34.38 | **61.72** |
| Flower102 | 52.73 | 27.80 | **60.30** |
| Purity(%) | | | |
| CCV | 25.46 | 28.60 | **29.57** |
| NonPlant | 14.15 | 69.78 | **75.29** |
| Flower17 | 47.65 | 35.10 | **63.60** |
| Flower102 | 38.76 | 21.25 | **53.07** |

We conduct experiments on computer vision datasets CCV [6], which contains 6773 YouTube videos over 20 semantic categories. We show our results in Table IV. The superiority experimental results of ours outperform existing SOTA kernel subspace clustering and even deep multi-view algorithm. It is noticed that the comparing method FMR in our paper is also a deep based strong baseline as [31], [57]. And [14] is considered to be a strong baseline for multiple kernel self-expressive subspace clustering. As can be seen, our method significantly outperforms theirs and the results clearly demonstrate the effectiveness of our proposed method.

[6]https://www.ee.columbia.edu/ln/dvmm/CCV/

### *F. Comparing with Existing Multi-view Ensemble Clustering*

We also conduct experiments comparing to existing multi-view ensemble clustering algorithms [8]–[10]. Multi-view ensemble clustering optimizes the optimal clustering partition matrix by aggregating a set of given pre-defined multiple partitions. As can be seen, our method significantly outperforms the competitors. We attribute the superiority with the following reasons:(i)more flexible similarity measure. Ensemble clustering jointly fuses multiple partitions to reach a consensus partition which heavily relies the quality of base partitions.(ii) capture nonlinear information. Kernel methods capture the nonlinear relationship with data items which is more practical in real applications.

TABLE V: The ACC NMI and Purity comparison of multi-view ensemble clustering methods and ours.

| Datasets | MVEC | M2VEC | Ours |
|---|---|---|---|
| ACC(%) | | | |
| Caltech-5 | 25.13 | 25.37 | 43.73 |
| Caltech-10 | 19.55 | 20.08 | 40.78 |
| Caltech-15 | 15.97 | 16.64 | 39.93 |
| Flower17 | 36.51 | 31.84 | 62.28 |
| YALE | 26.55 | 27.52 | 62.42 |
| AR10P | 20.62 | 20.15 | 56.15 |
| NMI(%) | | | |
| Caltech-5 | 61.34 | 62.47 | 73.69 |
| Caltech-10 | 49.61 | 50.24 | 66.90 |
| Caltech-15 | 42.06 | 43.88 | 63.01 |
| Flower17 | 40.19 | 36.80 | 61.72 |
| YALE | 36.43 | 34.90 | 63.48 |
| AR10P | 15.60 | 16.02 | 51.82 |
| Purity(%) | | | |
| Caltech-5 | 28.44 | 29.64 | 45.49 |
| Caltech-10 | 21.71 | 22.05 | 43.73 |
| Caltech-15 | 17.44 | 18.23 | 41.83 |
| Flower17 | 38.58 | 34.56 | 63.60 |
| YALE | 31.15 | 33.46 | 62.42 |
| AR10P | 21.46 | 22.54 | 56.15 |

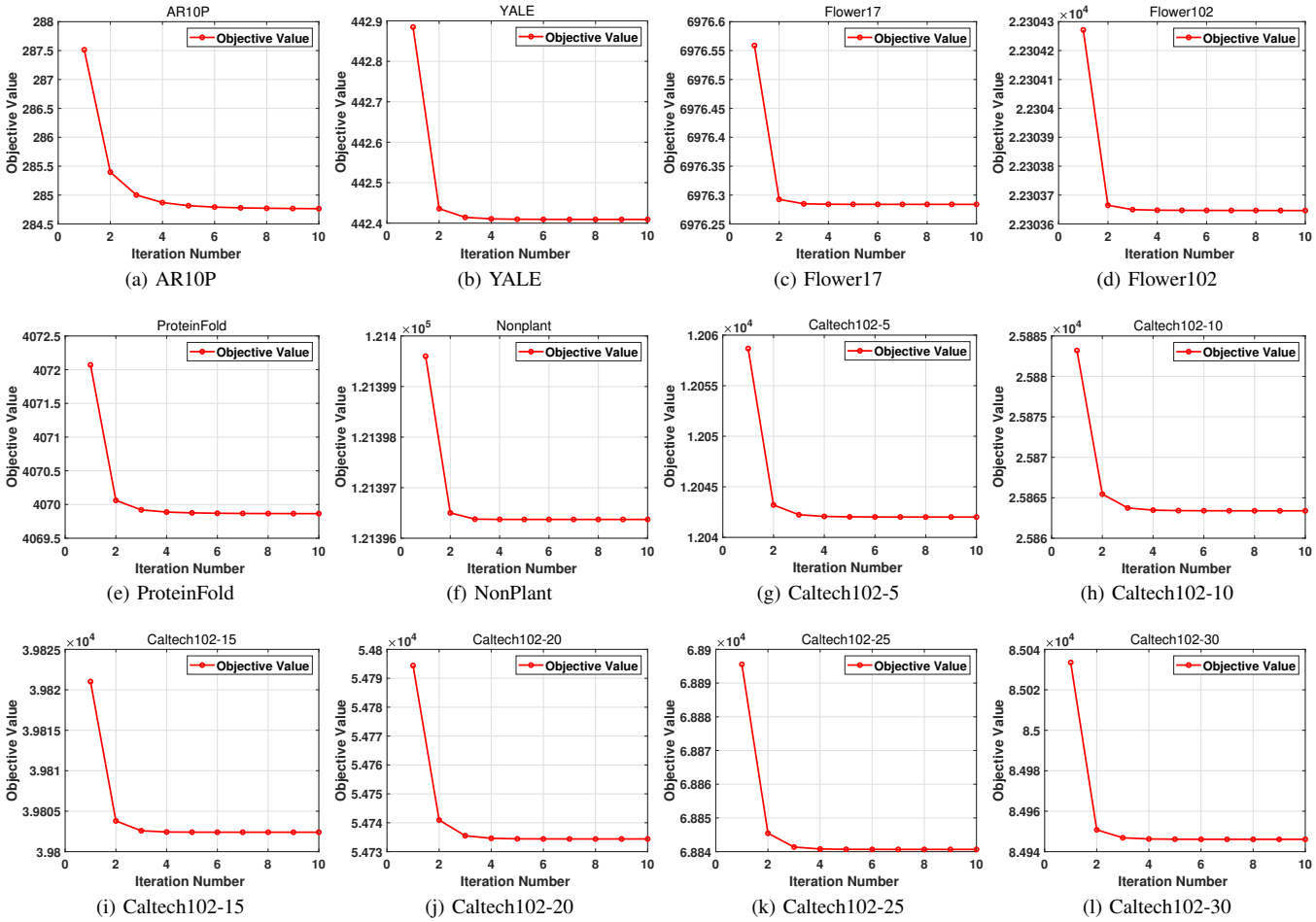
### *G. Running Time Comparison*

To compare the computational efficiency of the proposed algorithms, we record the running time of various algorithms on these benchmark datasets and report them in Table VI.

From this table, we have two aspects of observations. First, it can be observed that the time complexity of MKAM and FMR are relatively expensive over the other compared methods. Second, the proposed algorithm ranks second best in existing methods. Although MLFA achieves better in term of efficiency, the proposed method exceeds much better clustering performance as shown in Table III. Therefore, it is clear to see that the total computational cost of proposed method is less or much less than MKAM, FMR in our experiments. This is probably the main reason that our method is able to cost less time than the compared methods in most cases (as shown in Table VI).

### *H. Graph Refinement*

To directly illustrate the effectiveness of the proxy graph refinement on the late fusion base kernel partitions, we visualize the affinity matrix in Figure 4. As can be observed, our




(a) AR10P (b) YALE (c) Flower17 (d) Flower102

(e) ProteinFold (f) NonPlant (g) Caltech102-5 (h) Caltech102-10

(i) Caltech102-15 (j) Caltech102-20 (k) Caltech102-25 (l) Caltech102-30

Fig. 3: The convergence of the proposed LFMKC-PGR on the entire twelve datasets.

TABLE VI: The time comparison of representative MKC algorithms on three large benchmark datasets (in seconds).

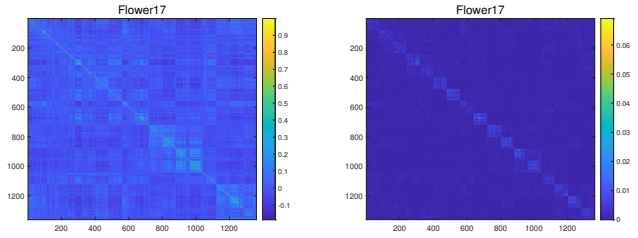
| Datasets | MKAM | MLFA | FMR | Proposed |
|---|---|---|---|---|
| Flower102 | 1027.4 | 90.71 | 1805.2 | 365.1 |
| Nonplant | 2694.29 | 21.18 | 464.8 | 148.0 |
| Caltech102-30 | 620.07 | 59.76 | 459.9 | 143.76 |

proposed method refines the base partitions and optimally be fused with the proxy graph. The noises in the affinity matrix shown in Figure 4a are eliminated and the clustering structure becomes clearer in Figure 4b. After adding constraint into the MKC optimization goal, the learned similarity graph is constructed in the kernel latent space rather than the original RKHS space.

We have also shown the t-sne visual results on the mfeat datasets in Figure 5 of the learned data representation on the 1-st, 3-rd, 5-th and 10-th iterations. The figures clearly show the separation of different clusters. Also, it can be observed form Figure 6, the learned affinity matrices show clearer block clustering structure with the variation of iterations.

### *I. Convergence and Parameter Sensitivity*

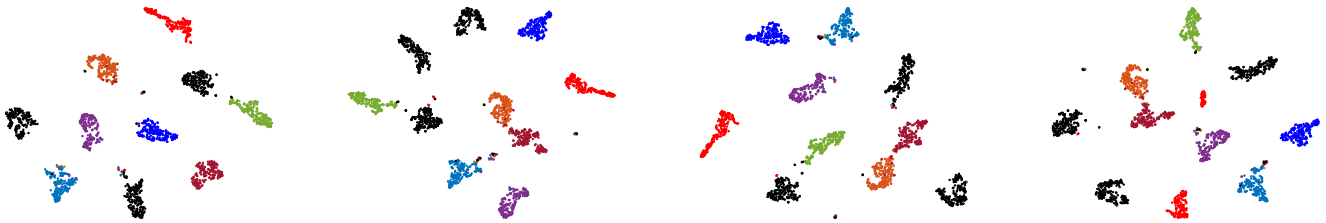
Our algorithm is theoretically guaranteed to converge to a local minimum according to [50]. We also conduct experiments to demonstrate the convergence of the proposed algorithm. The examples of the evolution of the objective value on the experimental results are shown in Figure 3. In the above experiments, we observe that the objective values of our algorithm monotonically decrease at each iteration. These results clearly verify our proposed algorithm's convergence.




(a) MLFA (b) Our Proposed

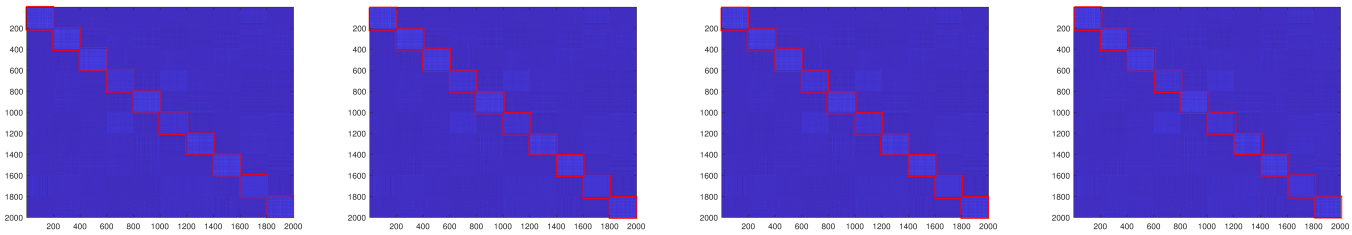
Fig. 4: An illustration of the learned affinity matrix on Flower17: (a) MLFA (b) Our Proposed LFMKC-PGR.

The Figure 7 shows an example of the sensitivity experimental results on AR10P and Flower17. From these figures, we observe that: i) LFMKC-PGR is practically stable against



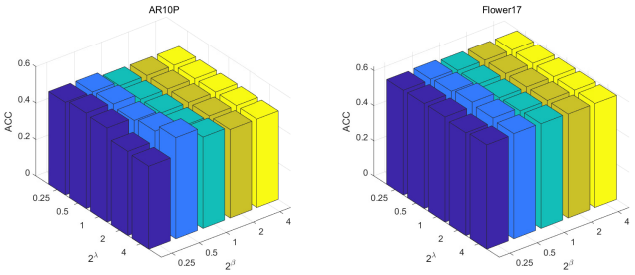
(a) 1-st iteration (b) 3-rd iteration (c) 5-th iteration (d) 10-th iteration

Fig. 5: An illustration of the learned data distribution with t-sne algorithm on mfeat datasets.



(a) 1-st iteration (b) 3-rd iteration (c) 5-th iteration (d) 10-th iteration

Fig. 6: An illustration of the learned affinity matrix on mfeat datasets.




(a) AR10P (b) Flower17

Fig. 7: The sensitivity of the proposed method with the variation of $\lambda$ and $\beta$ on benchmark datasets.

these parameters that it achieves competitive performance in a wide range of parameter settings; i) the ACC first increases to a high value and generally maintains it up to slight variation with values of two hypermeters. However, it still outperforms the second-best algorithm in most of the benchmarks.

## V. Conclusion

In this article, we propose a novel multiple kernel clustering method termed LFMKC-PGR which simultaneously optimize kernel base partitions and graph refinement. The kernel base partitions can be refined by the proposed proxy graph and negotiated with each other. Extensive experiments are conducted on twelve multi-kernel benchmark datasets, demonstrating the effectiveness of our proposed algorithm. In the future, we will consider how to preserve multi-view local information in the kernel partition space and further improve clustering performance.

## Acknowledgment

This work was supported by the National Key R&D Program of China (project no. 2020AAA0107100) and the National Natural Science Foundation of China (project no. 61922088, 61906020, 61825305, 62006237 and 61773392).

## References

[1] S. Yu, L. Tranchevent, X. Liu, W. Glanzel, J. A. Suykens, B. De Moor, and Y. Moreau, "Optimized data fusion for kernel k-means clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 5, pp. 1031–1039, 2012.

[2] M. Gönen and A. A. Margolin, "Localized data fusion for kernel k-means clustering with application to cancer biology," in *Advances in Neural Information Processing Systems*, 2014, pp. 1305–1313.

[3] X. Liu, Y. Dou, J. Yin, L. Wang, and E. Zhu, "Multiple kernel k - means clustering with matrix-induced regularization," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016, pp. 1888–1894.

[4] Y. Wang, W. Zhang, L. Wu, X. Lin, M. Fang, and S. Pan, "Iterative views agreement: an iterative low-rank based structured optimization method to multi-view spectral clustering," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, 2016, pp. 2153–2159.

[5] Y. Wang, L. Wu, X. Lin, and J. Gao, "Multiview spectral clustering via structured low-rank matrix factorization," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 10, pp. 4833–4843, 2018.

[6] Z. Kang, C. Peng, and Q. Cheng, "Twin learning for similarity and clustering: a unified kernel approach," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 2017, pp. 2080–2086.

[7] Y. Yang, Z. Ma, A. G. Hauptmann, and N. Sebe, "Feature selection for multimedia analysis by sharing information among multiple tasks," *IEEE Trans. Multim.*, vol. 15, no. 3, pp. 661–669, 2013.

[8] Z. Tao, H. Liu, S. Li, Z. Ding, and Y. Fu, "Marginalized multiview ensemble clustering," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 31, no. 2, pp. 600–611, 2020.

[9] H. Liu, M. Shao, S. Li, and Y. Fu, "Infinite ensemble clustering," *Data Min. Knowl. Discov.*, vol. 32, no. 2, pp. 385–416, 2018.

[10] Z. Tao, H. Liu, S. Li, Z. Ding, and Y. Fu, "From ensemble clustering to multi-view clustering," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, C. Sierra, Ed., 2017, pp. 2843–2849.

[11] Z. Zhang, L. Liu, F. Shen, H. T. Shen, and L. Shao, "Binary multi-view clustering," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 7, pp. 1774–1782, 2018.

[12] S. Zhou, E. Zhu, X. Liu, T. Zheng, Q. Liu, J. Xia, and J. Yin, "Subspace segmentation-based robust multiple kernel clustering," *Information Fusion*, vol. 53, pp. 145–154, 2020.

[13] Z. Ren, S. X. Yang, Q. Sun, and T. Wang, "Consensus affinity graph learning for multiple kernel clustering," *IEEE Transactions on Cybernetics*, 2020.

[14] Z. Ren and Q. Sun, "Simultaneous global and local graph structure preserving for multiple kernel clustering," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[15] 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 Transactions on Knowledge and Data Engineering*, 2020.

[16] J. Liu, X. Liu, J. Xiong, Q. Liao, S. Zhou, S. Wang, and Y. Yang, "Optimal neighborhood multiple kernel clustering with adaptive local kernels," *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[17] H.-J. Li, Z. Wang, J. Pei, J. Cao, and Y. Shi, "Optimal estimation of low-rank factors via feature level data fusion of multiplex signal systems," *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[18] X. Liu, X. Zhu, M. Li, L. Wang, E. Zhu, T. Liu, M. Kloft, D. Shen, J. Yin, and W. Gao, "Multiple kernel $k$-means with incomplete kernels," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 5, pp. 1191–1204, 2020.

[19] X. Liu, L. Liu, Q. Liao, S. Wang, Y. Zhang, W. Tu, C. Tang, J. Liu, and E. Zhu, "One pass late fusion multi-view clustering," in *International Conference on Machine Learning*. PMLR, 2021, pp. 6850–6859.

[20] X. Liu, L. Wang, X. Zhu, M. Li, E. Zhu, T. Liu, L. Liu, Y. Dou, and J. Yin, "Absent multiple kernel learning algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 6, pp. 1303–1316, 2020.

[21] J. Liu, F. Cao, X.-Z. Gao, L. Yu, and J. Liang, "A cluster-weighted kernel k-means method for multi-view clustering." in *AAAI*, 2020, pp. 4860–4867.

[22] X. Liu, X. Zhu, M. Li, L. Wang, C. Tang, J. Yin, D. Shen, H. Wang, and W. Gao, "Late fusion incomplete multi-view clustering," *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[23] C. M. Wong, C. M. Vong, P. K. Wong, and J. Cao, "Kernel-based multilayer extreme learning machines for representation learning," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 3, pp. 757–762, 2016.

[24] S. Wang, X. Liu, E. Zhu, C. Tang, J. Liu, J. Hu, J. Xia, and J. Yin, "Multi-view clustering via late fusion alignment maximization." in *IJCAI*, 2019, pp. 3778–3784.

[25] X. Liu, M. Li, C. Tang, J. Xia, J. Xiong, L. Liu, M. Kloft, and E. Zhu, "Efficient and effective regularized incomplete multi-view clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[26] Z. Kang, X. Zhao, C. Peng, H. Zhu, J. T. Zhou, X. Peng, W. Chen, and Z. Xu, "Partition level multiview subspace clustering," *Neural Networks*, vol. 122, pp. 279–288, 2020.

[27] X. Peng, C. Lu, Z. Yi, and H. Tang, "Connections between nuclear-norm and frobenius-norm-based representations," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 1, pp. 218–224, 2016.

[28] M. Sun, S. Wang, P. Zhang, X. Liu, S. Zhou, X. Guo, and E. Zhu, "Projective multiple kernel subspace clustering," *IEEE Transactions on Multimedia*, 2021.

[29] M. Chen, L. Huang, C.-D. Wang, and D. Huang, "Multi-view clustering in latent embedding space." in *AAAI*, 2020, pp. 3513–3520.

[30] X. Li, H. Zhang, R. Wang, and F. Nie, "Multi-view clustering: A scalable and parameter-free bipartite graph fusion method," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[31] R. Li, C. Zhang, H. Fu, X. Peng, T. Zhou, and Q. Hu, "Reciprocal multi-layer subspace learning for multi-view clustering," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 8172–8180.

[32] Z. Kang, X. Lu, J. Yi, and Z. Xu, "Self-weighted multiple kernel learning for graph-based clustering and semi-supervised classification," *arXiv preprint arXiv:1806.07697*, 2018.

[33] M. Li, X. Liu, L. Wang, Y. Dou, J. Yin, and E. Zhu, "Multiple kernel clustering with local kernel alignment maximization," in *IJCAI*, 2016.

[34] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in neural information processing systems*, 2002, pp. 849–856.

[35] I. S. Dhillon, Y. Guan, and B. Kulis, "Kernel k-means: spectral clustering and normalized cuts," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2004, pp. 551–556.

[36] E. Elhamifar and R. Vidal, "Sparse subspace clustering," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 2790–2797.

[37] S. Luo, C. Zhang, W. Zhang, and X. Cao, "Consistent and specific multi-view subspace clustering," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[38] C.-D. Wang, J.-H. Lai, and S. Y. Philip, "Multi-view clustering based on belief propagation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 4, pp. 1007–1021, 2015.

[39] Z. Ren, Q. Sun, B. Wu, X. Zhang, and W. Yan, "Learning latent low-rank and sparse embedding for robust image feature extraction," *IEEE Transactions on Image Processing*, vol. 29, no. 1, pp. 2094–2107, 2019.

[40] D. Huang, C.-D. Wang, J.-S. Wu, J.-H. Lai, and C.-K. Kwoh, "Ultra-scalable spectral clustering and ensemble clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 6, pp. 1212–1226, 2019.

[41] X. Zhu, S. Zhang, R. Hu, Y. Zhu *et al.*, "Local and global structure preservation for robust unsupervised spectral feature selection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 3, pp. 517–529, 2017.

[42] X. Zhu, S. Zhang, W. He, R. Hu, C. Lei, and P. Zhu, "One-step multi-view spectral clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 10, pp. 2022–2034, 2018.

[43] X. Peng, H. Tang, L. Zhang, Z. Yi, and S. Xiao, "A unified framework for representation-based subspace clustering of out-of-sample and large-scale data," *IEEE transactions on neural networks and learning systems*, vol. 27, no. 12, pp. 2499–2512, 2015.

[44] K. Zhan, C. Niu, C. Chen, F. Nie, C. Zhang, and Y. Yang, "Graph structure fusion for multiview clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 10, pp. 1984–1993, 2018.

[45] S. Xiang, F. Nie, C. Zhang, and C. Zhang, "Nonlinear dimensionality reduction with local spline embedding," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1285–1298, 2008.

[46] Q. Wang, Z. Qin, F. Nie, and X. Li, "Spectral embedded adaptive neighbors clustering," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 4, pp. 1265–1271, 2018.

[47] V. M. Patel, H. Van Nguyen, and R. Vidal, "Latent space sparse subspace clustering," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 225–232.

[48] V. M. Patel and R. Vidal, "Kernel sparse subspace clustering," in *2014 IEEE international conference on image processing (ICIP)*. IEEE, 2014, pp. 2849–2853.

[49] M. Yin, Y. Guo, J. Gao, Z. He, and S. Xie, "Kernel sparse subspace clustering on symmetric positive definite manifolds," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5157–5164.

[50] J. C. Bezdek and R. J. Hathaway, "Convergence of alternating optimization," *Neural, Parallel Sci. Comput.*, vol. 11, no. 4, pp. 351–368, 2003.

[51] H.-C. Huang, Y.-Y. Chuang, and C.-S. Chen, "Multiple kernel fuzzy clustering," *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 1, pp. 120–134, 2012.

[52] A. Kumar, P. Rai, and H. Daume, "Co-regularized multi-view spectral clustering," in *Advances in neural information processing systems*, 2011, pp. 1413–1421.

[53] L. Du, P. Zhou, L. Shi, H. Wang, M. Fan, W. Wang, and Y.-D. Shen, "Robust multiple kernel k-means using l21-norm," in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[54] R. Xia, Y. Pan, L. Du, and J. Yin, "Robust multi-view spectral clustering via low-rank and sparse decomposition," in *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[55] R. Li, C. Zhang, Q. Hu, P. Zhu, and Z. Wang, "Flexible multi-view representation learning for subspace clustering." in *IJCAI*, 2019, pp. 2916–2922.

[56] Z. Li, Q. Wang, Z. Tao, Q. Gao, and Z. Yang, "Deep adversarial multi-view clustering network." in *IJCAI*, 2019, pp. 2952–2958.
[57] R. Wang, F. Nie, Z. Wang, H. Hu, and X. Li, "Parameter-free weighted multi-view projected clustering with structured graph learning," *IEEE Transactions on Knowledge and Data Engineering*, 2019.

**Siwei Wang** is pursuing his P.H.D degree in National University of Defense Technology (NUDT), China. His current research interests include kernel learning, unsupervised multiple-view learning, scalable clustering and deep unsupervised learning. He has published several papers and served as PC member/ Reviewer in top journals and conferences such as IEEE TKDE/TNNLS/TIP/TCYB/TMM, ICML/CVPR/ECCV/ICCV/AAAI/IJCAI, etc.

**Xinwang Liu** received his PhD degree from National University of Defense Technology (NUDT), China. He is now full professor of School of Computer, NUDT. His current research interests include kernel learning and unsupervised feature learning. Dr. Liu has published 60+ peer-reviewed papers, including those in highly regarded journals and conferences such as IEEE T-PAMI, IEEE T-KDE, IEEE T-IP, IEEE T-NNLS, IEEE T-MM, IEEE T-IFS, ICML, NeurIPS, ICCV, CVPR, AAAI, IJCAI, etc. He is a senior member of IEEE. More information can be found at https://xinwangliu.github.io.

**Li Liu** received the BSc degree in communication engineering, the MSc degree in photogrammetry and remote sensing and the Ph.D. degree in information and communication engineering from the National University of Defense Technology (NUDT), China, in 2003, 2005 and 2012, respectively. She joined the faculty at NUDT in 2012, where she is currently an Associate Professor with the College of System Engineering. She was a cochair of seven International Workshops at CVPR, ICCV, and ECCV. She is going to lecture a tutorial at CVPR'19. She was a guest editor of special issues for IEEE TPAMI and IJCV. Her current research interests include facial behavior analysis, texture analysis, image classification, object detection and recognition. Her papers have currently over 1800 citations in Google Scholar. She currently serves as Associate Editor of the Visual Computer Journal.

**Sihang Zhou** received his PhD degree in computer science from National University of Defense Technology (NUDT), China in 2019. He received his M.S. degree in computer science from the same school in 2014 and his bachelor's degree in information and computing science from the University of Electronic Science and Technology of China (UESTC) in 2012. He is now a lecturer of College of Intelligence Science and Technology, NUDT. His current research interests include machine learning, pattern recognition and medical image analysis.

**En Zhu** received his PhD degree from National University of Defense Technology (NUDT), China. He is now Professor at School of Computer Science, NUDT, China. His main research interests are pattern recognition, image processing, machine vision and machine learning. Dr. Zhu has published 60+ peer-reviewed papers, including IEEE T-CSVT, IEEE T-NNLS, PR, AAAI, IJCAI, etc. He was awarded China National Excellence Doctoral Dissertation.