Multiple Kernel Clustering With Adaptive Multi-Scale Partition Selection

Jun Wang, Zhenglai Li, Chang Tang, *Senior Member, IEEE*, Suyuan Liu, Xinhang Wan, and Xinwang Liu, *Senior Member, IEEE*

***Abstract*—Multiple kernel clustering (MKC) enhances clustering performance by deriving a consensus partition or graph from a predefined set of kernels. Despite many advanced MKC methods proposed in recent years, the prevalent approaches involve incorporating all kernels by default to capture diverse information within the data. However, learning from all kernels may not be better than one of a few kernels, particularly since some kernels exhibit a higher proportion of noise than semantic content. Additionally, existing MKC methods, whether based on early-fusion or late-fusion approaches, predominantly rely on pairwise relationships among samples or cluster structures, neglecting potential correlations between these two aspects. To this end, we propose a multiple kernel clustering with an adaptive multi-scale partition selection method (MPS), which exploits multiple-dimensional representations and the pairwise cluster structure for clustering. By the proposed kernel selection framework, potentially harmful kernels are dynamically excluded during the kernel fusion process, and then the multi-scale partitions and similarity graphs derived from the retained kernels are utilized to facilitate the improved consensus partition generation. Finally, extensive experiments are conducted to demonstrate the effectiveness of MPS on eight benchmark datasets.**

***Index Terms*—Kernel selection, multi-scale embedding learning, multiple kernel clustering, partition fusion.**

I. Introduction

CLUSTERING, as a fundamental technique for uncovering latent patterns in data through unsupervised analysis, has garnered widespread attention across various domains, including natural language processing (NLP) [1], [2], [3], [4], [5], object detection [6], [7], [8], [9], hyperspectral image processing [10], [11], [12], [13], and single cell clustering [14], [15]. However, as data types become increasingly diverse, object data often exhibits a multitude of heterogeneous features. For instance, individuals may hold accounts in multiple banks, or patients may undergo various medical checkups within a hospital setting, which can be categorized as multi-view data. In practical terms, employing traditional single clustering methods directly on these complex data sets can prove challenging [16]. Consequently, there has been a surge in the development of advanced multi-view clustering (MVC) methods in recent years.

Among the array of existing MVC methods, graph-based clustering stands out as a widely explored approach. This method is designed to construct a unified similarity graph (a.k.a affinity graph) from the original multi-view data [17], [18], [19], [20], [21]. In this way, the ultimate clustering performance is intrinsically tied to the quality of this consensus similarity graph. Moreover, given that multi-view data is not invariably linearly separable, the resulting similarity graph based on metrics like Euclidean distance may struggle to well reveal the underlying data structure [22], [23]. Consequently, certain methods introduce kernel clustering to address non-linearly separable data by mapping it into a high-dimensional feature space—this methodology is commonly referred to as multiple kernel clustering (MKC).

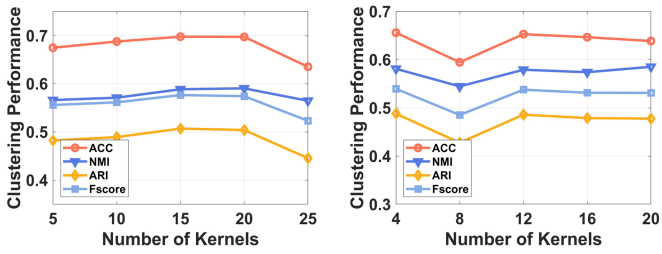
MKC primarily employs multiple kernel functions to generate one or more kernels for each view, optimizing them through a linearly weighted approach for downstream tasks [24], [25], [26]. Based on the stage of feature fusion, current approaches predominantly fall into two categories: early-fusion [27], [28] and late-fusion [29], [30] based methods. Early-fusion strategies aim to construct a unified kernel matrix from a series of predefined base kernels, which principally capture the correlations among samples. This approach effectively leverages the original data information by exploiting the pairwise structural relationships among samples, such as [22], [31]. Conversely, late-fusion techniques focus on constructing base partition matrices from individual kernel matrices, subsequently integrating these into a consensus one. This method emphasizes the cluster structure between samples and clusters, improving clustering performances by treating each base partition as an embedding that encodes clustering information [25], [32], [33]. Additionally, as a prominent domain within machine learning, deep learning methods have garnered significant achievements across various research domains. Notably, in contrast to multiple kernel learning methods, deep learning-based models stand out for their utilization of hierarchical frameworks that stack numerous nonlinear models to generate latent data representations. This characteristic distinguishes them from traditional multiple kernel learning, which can be viewed as a shallow special case. Leveraging the potent representation capabilities inherent to deep learning, several advanced deep multiple kernel clustering

Manuscript received 9 January 2024; revised 12 April 2024; accepted 4 May 2024. Date of publication 13 May 2024; date of current version 27 September 2024. This work was supported by the National Natural Science Foundation of China (NO. 62325604, 62276271). Recommended for acceptance by X. He. *(Corresponding author: Chang Tang; Xinwang Liu)*

Jun Wang, Suyuan Liu, Xinhang Wan, and Xinwang Liu are with the School of Computer, National University of Defense Technology, Changsha 410073, China (e-mail: wang_jun@nudt.edu.cn; suyuanliu@nudt.edu.cn; wanxinhang@nudt.edu.cn; xinwangliu@nudt.edu.cn).

Zhenglai Li and Chang Tang are with the School of Computer Science, China University of Geosciences, Wuhan 430074, China (e-mail: yuezhenguan@cug.edu.cn; tangchang@cug.edu.cn).

Digital Object Identifier 10.1109/TKDE.2024.3399738



(a) MSRC_v1 (b) EMNIST

Fig. 1. Clustering performance using different numbers of kernels on MSRC_v1 and EMNIST dataset.

techniques have been proposed, such as [34], [35], [36], [37]. For instance, [38] introduced a subspace clustering penalty into the architecture of autoencoders, where the optimization process alternates between refining the self-representation matrix and employing stochastic gradient descent. Furthermore, [39] incorporated KL-divergence to derive cluster assignments from the produced latent embedding. However, training deep models often entails intricate nonlinear optimizations, and the detailed design of network architectures lacks clear theoretical guidance.

While the above methods have demonstrated commendable clustering performance in recent years, certain limitations persist. Firstly, many MKC methods predominantly focus on the fusion of all kernels, overlooking the fact that an increase in the number of kernels does not necessarily lead to an improvement in clustering performance. Taking the multiple kernel k-means as an example [29], as depicted in Fig. 1, the final performance does not exhibit a consistent improvement with the addition of more kernels, indicating instances where some kernels contribute negatively to information gain. Secondly, the conventional approach to obtain discrete clustering results involves directly applying k-means on the consensus partition matrix with the size of $n \times k$, where n and k represent the number of samples and partition dimensions, respectively. This method relies heavily on the consensus partition matrix, which is generated by fusing all independent base partitions. However, this approach encounters an information bottleneck in the final clustering process due to the loss of information resulting from the sudden dimension drop [40]. Lastly, from the perspective of utilizing original data information, early-fusion-based MKC methods primarily leverage pairwise structures among samples for clustering, while late-fusion-based MKC methods predominantly utilize the cluster structure between samples and clusters. Unfortunately, existing methods neglect potential correlations between these two aspects [41].

To tackle the aforementioned challenges, we introduce a method for multiple kernel clustering with adaptive multi-scale partition selection. This method comprises two key steps: kernel selection and kernel fusion, as illustrated in the detailed flowchart in Fig. 2. Initially, the original data undergoes mapping into diverse kernel matrices using various kernel functions. Diverging from conventional MKC methods, the proposed method generates multi-scale base partitions for each kernel at the feature level, concurrently constructing a corresponding similarity graph at the structural level. This facilitates the effective distillation and preservation of both pairwise information among samples and cluster information. Considering that real-world data often contains significant redundancy, the direct fusion of all kernel matrices into a unified one may negatively impact the ultimate clustering performance. To this end, we employ a straightforward yet effective kernel selection strategy in our proposed method to identify kernels of relatively high quality. Subsequently, these selected kernels are employed to construct a consensus one for the ultimate clustering task. Notably, owing to the inherent challenge of precisely evaluating the quality of each kernel, we integrate the processes of kernel selection and fusion into a unified framework, allowing for dynamic and adaptive selection of kernels. Finally, we design an efficient iterative optimization algorithm to solve the proposed model.

In summary, this paper delineates its contributions as follows.

- We develop a novel dynamic adaptive kernel selection strategy for multiple kernel clustering. This subtle adjustment in the kernel selection process yields an enhancement in the quality of the generated consensus kernel, thereby enhancing the overall clustering performance.
- We propose a novel method that integrates multi-scale partitioning and similarity graph fusion to effectively capture both pairwise feature information and structural properties concurrently. Furthermore, we incorporate this kernel fusion approach along with the previously outlined kernel selection strategy into a unified framework, facilitating mutual reinforcement between the two techniques.
- We design an alternating optimization algorithm to address the resultant model, backed by a comprehensive set of experiments conducted on eight benchmark datasets, validating the efficacy of the proposed method.

The subsequent sections of this paper are organized as follows. Section II provides a brief introduction to both multiple kernel k-means and adaptive neighbor graph learning. In Section III, we delve into the proposed method, offering comprehensive details encompassing its motivation and the process of constructing the objective function. The specific optimization procedures are presented in Section IV. To assess the efficacy of the proposed method across real datasets, extensive experiments are conducted and the corresponding results are reported in Section V. Finally, the paper is concluded in Section VI.

II. Related Work

A. Multiple Kernel k-Means

As one of the classical clustering methods, kernel k-means aims to group the original data $\mathbf{X} \in \mathbb{R}^{n \times d}$ into k classes based on a specific clustering loss, where d denotes the number of dimensions [42]. Let $\varphi(\cdot) : x \in \mathcal{X} \mapsto \mathcal{H}$ denote the mapping function that projects the samples into a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$, i.e., $\phi_i = \varphi(x_i)$ [43]. The objective function of kernel k-means clustering can then be formulated as [44]:

$$\min_{\mathbf{H}} \mathrm{Tr}(\mathbf{K}(\mathbf{I}_n - \mathbf{H}\mathbf{H}^\top)), \quad s.t. \, \mathbf{H}^\top \mathbf{H} = \mathbf{I}_k. \tag{1}$$

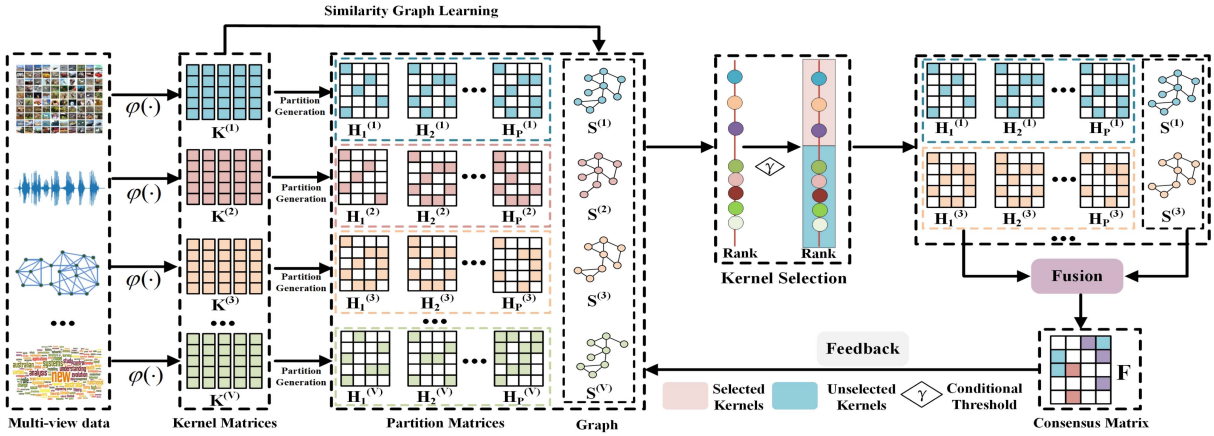


Fig. 2. Framework of MPS. Firstly, the original multi-view data is transformed into corresponding kernel matrices $\{\mathbf{K}^{(i)}\}_{i=1}^{V}$ using diverse kernel functions $\varphi(\cdot)$. Subsequently, multi-scale partitions and similarity graphs are constructed for each kernel. To ensure that the selected kernels contribute positively to the resulting consensus partition, we calculate the weights of each kernel and rank them, then choose higher-quality kernels based on this ranking. Ultimately, the consensus partition is generated through the fusion of the selected kernels.

where $\mathbf{K}(x_i, x_j) = \phi_i^\top \phi_j$. The optimal solution of $\mathbf{H}$ can be obtained by conducting singular value decomposition (SVD) on the kernel matrix $\mathbf{K}$. When the base partition matrix $\mathbf{H}$ is generated, the ultimate clustering results can be obtained by executing standard k-means clustering on it.

As observed from (1), the final clustering performance is heavily dependent on the quality of the constructed kernel matrix. However, selecting the optimal kernel proves challenging in practical scenarios. To this end, various methods for multiple kernel k-means clustering have been developed [45], [46], [47], [48]. Specifically, considering a series of base kernel matrices $\{\mathbf{K}^{(i)}\}_{i=1}^{V}$, where V is the total number of kernels, the multiple kernel k-means clustering method assumes that the consensus kernel $\mathbf{K}_\sigma$ is generated by assigning weights to them in a linear combination, and thereby the objective function of this method can be expressed as follows:

$$\begin{aligned} &\min_{\mathbf{H},\boldsymbol{\sigma}} \mathrm{Tr}(\mathbf{K}_\sigma(\mathbf{I}_n - \mathbf{H}\mathbf{H}^\top)), \\ &s.t.\, \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k, \sigma^\top \mathbf{1} = 1, \sigma_i \geq 0, \forall i. \end{aligned} \tag{2}$$

where $\mathbf{K}_\sigma = \sum_{i=1}^{V} \sigma_i^2 \mathbf{K}^{(i)}$, and σ_i denotes the weight of i-th kernel.

B. Adaptive Neighbor Graph Learning

In the existing clustering methods, the graph-based learning methods have achieved satisfying performance due to their superiority in exploring diverse data structures. In order to obtain the high-quality similarity graph for the subsequent clustering task, an adaptive graph learning method is proposed in [49], the central idea of which is that two samples that are closer together should be assigned a larger weight. Thus, the corresponding objective function is constructed as follows:

$$\min_{\mathbf{S}} \sum_{i,j=1}^{n} \|x_i - x_j\|_2^2 s_{ij} + \lambda s_{ij}^2, s.t.\, \mathbf{s}_i^\top \mathbf{1} = 1, \mathbf{0} \leq \mathbf{s}_i \leq \mathbf{1}. \tag{3}$$

where s_{ij} denotes the similarity between samples i and j. λ is a trade-off parameter, which can be tuned according to the number of neighbors, and the detailed optimization process can refer to [49]. On the basis of (3), many advanced multi-view clustering methods are proposed, such as [50], [51], [52], [53].

III. Proposed Method

A. Kernel Selection and Fusion

In existing early-fusion and late-fusion-based MKC methods, despite variations in their formulations, they uniformly construct a consensus partition of size $n \times k$ for the final clustering. This partition comprises eigenvectors corresponding to the first k largest eigenvalues of the kernel matrix. According to [54], eigenvectors associated with larger eigenvalues contain more information. Thus, an optimal scenario for MKC is one in which the k-dimensional eigenvectors encapsulate the majority of information in the kernel matrix. However, as noted in [40], directly fusing the k-dimensional base partitions is not an ideal choice, thereby presenting a critical challenge for MKC methods: how to generate a high-quality consensus partition from the original kernels?

In response to this challenge, we propose a multi-scale partition fusion strategy. Diverging from previous methods, we construct multiple partition matrices of varying dimensions using (1), such as $2\,k, 3\,k$, and so forth, instead of a singular k-dimensional matrix. Subsequently, a consensus partition is generated by adaptively fusing these multi-scale partitions. The specific process can be formulated as follows:

$$\begin{aligned} &\min_{\mathbf{F},\boldsymbol{\beta}} \sum_{i=1}^{V}\sum_{j=1}^{P} \|\mathbf{F}\mathbf{F}^\top - \beta_j \mathbf{H}_j^{(i)} \mathbf{H}_j^{(i)\top}\|_F^2, \\ &s.t.\, \mathbf{F}^\top\mathbf{F} = \mathbf{I}, \boldsymbol{\beta} \geq \mathbf{0}, \boldsymbol{\beta}^\top \boldsymbol{\beta} = 1. \end{aligned} \tag{4}$$

where $\mathbf{F}$ and $\mathbf{H}_j^{(i)}$ denote the consensus partition matrix and the j-th scale partition for the i-th kernel, respectively. β_j denotes the j-th scale weight.

For (4), the primary objective is to effectively distill and preserve semantic information from each kernel. In practical applications, the original data inevitably contains noise, leading to kernels that are tainted with both semantic information and irrelevant noise. Some kernels may even exhibit a higher proportion of noise than semantic content, resulting in negative information gain. Utilizing multi-scale partition matrices derived from these deleterious kernels to construct the consensus matrix can detrimentally impact clustering performance. To mitigate this issue, we introduce a dynamic adaptive kernel selection strategy to identify and exclude potentially harmful kernels. The strategy is delineated as follows:

$$\min_{\mathbf{F},\boldsymbol{\beta}} \sum_{i=1}^{V}\sum_{j=1}^{P} \gamma_i \|\mathbf{F}\mathbf{F}^\top - \beta_j \mathbf{H}_j^{(i)}\mathbf{H}_j^{(i)\top}\|_F^2,$$
$$s.t.\ \mathbf{F}^\top\mathbf{F} = \mathbf{I}, \boldsymbol{\gamma} \in \{0,1\}^V, \|\boldsymbol{\gamma}\|_0 = M, \boldsymbol{\beta} \geq \mathbf{0}, \boldsymbol{\beta}^\top\boldsymbol{\beta} = 1. \tag{5}$$

where γ_i is a zero-one indicator variable, which signifies whether a kernel matrix is incorporated into the proposed modal, while M represents the total number of selected kernels. By preserving kernels that contain a higher volume of informative content, our proposed method can obtain an improved consensus partition, as it remains unaffected by the perturbations introduced by lower-quality partitions.

B. Pairwise Structure Learning

Although a robust feature representation $\mathbf{F}$ can be derived from the above (5), we find that it mainly depicts the affiliation association between the samples and clusters, ignoring the potential pairwise relationships among different samples. In the experimental results reported in [41], the structural information among samples also plays a crucial role in improving the final clustering performance. To capture this prior property, we advocate constructing a similarity graph from each kernel to enhance the quality of the generated partition matrix. In particular, building upon (3), we reformulate it in kernel form to establish the corresponding similarity graph for each kernel:

$$\min_{\mathbf{S}} \sum_{i,j=1}^{n} \|\phi(x_i) - \phi(x_j)\|_2^2 s_{ij} + \lambda s_{ij}^2.$$
$$s.t.\ \mathbf{s}_i^\top \mathbf{1} = 1, \mathbf{0} \leq \mathbf{s}_i \leq \mathbf{1}. \tag{6}$$

Since λ is practically determined by the number of neighbors in the k-nearest neighbors ($k$nn) graph, it becomes dispensable when the number of k-nn neighbors is fixed. Each base partition can be regarded as a specific form of feature representation for the samples. Once the similarity graph is constructed from the original data in the kernel space, a reasonable assumption is that the similarity among samples within the consensus partition should align with the original feature representation. Hence, the formulation is derived as follows:

$$\min_{\mathbf{F}} \sum_{i,j=1}^{n} \|f_i - f_j\|_2^2 s_{ij}^{(v)} \Leftrightarrow \min_{\mathbf{F}} \|\mathbf{F}\mathbf{F}^\top - \mathbf{S}^{(v)}\|_F^2,$$
$$s.t.\ \mathbf{F}^\top\mathbf{F} = \mathbf{I}. \tag{7}$$

where $\mathbf{S}^{(v)}$ represents the similarity graph of v-th kernel.

C. Overall Objective Function

Based on the preceding discussions, the objective function can be explicitly formulated by combining the (5) and (7), i.e.:

$$\min_{\mathbf{F},\boldsymbol{\gamma},\boldsymbol{\beta}} \alpha \sum_{i=1}^{V}\sum_{j=1}^{P} \gamma_i \|\mathbf{F}\mathbf{F}^\top - \beta_j \mathbf{H}_j^{(i)}\mathbf{H}_j^{(i)\top}\|_F^2$$
$$+(1-\alpha)\sum_{i=1}^{V} \gamma_i \|\mathbf{F}\mathbf{F}^\top - \mathbf{S}^{(i)}\|_F^2.$$
$$s.t.\ \mathbf{F}^\top\mathbf{F} = \mathbf{I}, \boldsymbol{\gamma} \in \{0,1\}^V, \|\boldsymbol{\gamma}\|_0 = M, \boldsymbol{\beta} \geq \mathbf{0}, \boldsymbol{\beta}^\top\boldsymbol{\beta} = 1. \tag{8}$$

Upon initial inspection, it is apparent that the provided function solely addresses kernel selection weights, neglecting the consideration of smooth weights associated with the selected kernels—a deviation from the typical approach in multiple kernel clustering. To rectify this, our proposed method incorporates explicit kernel weights. Specifically, we link these kernel weights to the objective function, wherein smaller loss values correspond to larger weights assigned to the respective kernels. Drawing inspiration from the method presented in [55], we formulate the final objective function as follows:

$$\min_{\mathbf{F},\boldsymbol{\gamma},\boldsymbol{\beta}} \alpha \sum_{i=1}^{V}\sum_{j=1}^{P} \gamma_i \|\mathbf{F}\mathbf{F}^\top - \beta_j \mathbf{H}_j^{(i)}\mathbf{H}_j^{(i)\top}\|_F$$
$$+(1-\alpha)\sum_{i=1}^{V} \gamma_i \|\mathbf{F}\mathbf{F}^\top - \mathbf{S}^{(i)}\|_F.$$
$$s.t.\ \mathbf{F}^\top\mathbf{F} = \mathbf{I}, \boldsymbol{\gamma} \in \{0,1\}^V, \|\boldsymbol{\gamma}\|_0 = M, \boldsymbol{\beta} \geq \mathbf{0}, \boldsymbol{\beta}^\top\boldsymbol{\beta} = 1. \tag{9}$$

In (9), the absence of explicitly provided weights for the selected kernels may appear unconventional. To address this, we elaborate on the determination of weights for the selected kernels through Theorem 1.

Theorem 1. Suppose the selected kernels are obtained, (9) can be rewritten as:

$$\min_{\mathbf{F},\boldsymbol{\beta}} \alpha \sum_{i=1}^{M}\sum_{j=1}^{P} \|\mathbf{F}\mathbf{F}^\top - \beta_j \mathbf{H}_j^{(i)}\mathbf{H}_j^{(i)\top}\|_F$$
$$+ (1-\alpha)\sum_{i=1}^{M} \|\mathbf{F}\mathbf{F}^\top - \mathbf{S}^{(i)}\|_F,$$
$$s.t.\ \mathbf{F}^\top\mathbf{F} = \mathbf{I}, \boldsymbol{\beta} \geq \mathbf{0}, \boldsymbol{\beta}^\top\boldsymbol{\beta} = 1. \tag{10}$$

then (10) is equivalent to:

$$\begin{aligned}&\min_{\mathbf{F},\boldsymbol{\mu},\boldsymbol{\omega},\boldsymbol{\beta}} \alpha\sum_{i=1}^{M}\sum_{j=1}^{P}\mu_i\|\mathbf{F}\mathbf{F}^\top-\beta_j\mathbf{H}_j^{(i)}\mathbf{H}_j^{(i)\top}\|_F^2\\&+(1-\alpha)\sum_{i=1}^{M}\omega_i\|\mathbf{F}\mathbf{F}^\top-\mathbf{S}^{(i)}\|_F^2,\\&s.t.\,\mathbf{F}^\top\mathbf{F}=\mathbf{I},\boldsymbol{\beta}\geq\mathbf{0},\boldsymbol{\beta}^\top\boldsymbol{\beta}=1,\\&\boldsymbol{\mu}\geq\mathbf{0},\boldsymbol{\mu}^\top\boldsymbol{\mu}=1,\boldsymbol{\omega}\geq\mathbf{0},\boldsymbol{\omega}^\top\boldsymbol{\omega}=1.\end{aligned}\tag{11}$$

Proof. Let $\mathcal{L}_1^{(i)}=\sum_{j=1}^{P}\|\mathbf{F}\mathbf{F}^\top-\beta_j\mathbf{H}_j^{(i)}\mathbf{H}_j^{(i)\top}\|_F^2$, and $\mathcal{L}_2^{(i)}=\|\mathbf{F}\mathbf{F}^\top-\mathbf{S}^{(i)}\|_F^2$, then the Lagrange function of (10) is:

$$\alpha\sum_{i=1}^{M}\mathcal{L}_1^{(i)\frac{1}{2}}+(1-\alpha)\sum_{i=1}^{M}\mathcal{L}_2^{(i)\frac{1}{2}}+\Delta,\tag{12}$$

where Δ denotes the Lagrange multiplier. When taking the derivative of (12) with respect to $\mathbf{F}$ and setting its value to zero, we can get:

$$\alpha\sum_{i=1}^{M}\mu_i\frac{\partial\mathcal{L}_1^{(i)}}{\partial\mathbf{F}}+(1-\alpha)\sum_{i=1}^{M}\omega_i\frac{\partial\mathcal{L}_2^{(i)}}{\partial\mathbf{F}}+\frac{\partial\Delta}{\partial\mathbf{F}}=0,\tag{13}$$

where

$$\mu_i=\frac{\mathcal{L}_1^{(i)(-\frac{1}{2})}}{2},\omega_i=\frac{\mathcal{L}_2^{(i)(-\frac{1}{2})}}{2}.\tag{14}$$

If μ_i and ω_i are fixed, then the derivation of Lagrange function with respect to $\mathbf{F}$ in (11) is equal to (13), and thereby the (10) is equivalent to (11). In this way, the weights of selected base partitions and similarity graphs, i.e., μ_i and ω_i, can be adaptively determined, respectively. □

IV. Optimization

In this section, a three-fold iterative optimization algorithm is designed to solve the resultant problem (9). The detailed solving processes are presented in the following.

Update $\boldsymbol{\gamma}$: When $\mathbf{F}$ and $\boldsymbol{\beta}$ are fixed, the (9) can be expressed as:

$$\min_{\boldsymbol{\gamma}}\sum_{i=1}^{V}\gamma_i\mathcal{L}^{(i)},\quad s.t.\,\boldsymbol{\gamma}\in\{0,1\}^V,\|\boldsymbol{\gamma}\|_0=M.\tag{15}$$

in which

$$\mathcal{L}^{(i)}=\alpha\mathcal{L}_1^{(i)\frac{1}{2}}+(1-\alpha)\mathcal{L}_2^{(i)\frac{1}{2}}.\tag{16}$$

For (15), it can be solved as follows:

$$\gamma_i=\begin{cases}1, & if\ \mathcal{L}^{(i)}\leq\mathcal{L}_{[M]},\\0, & Otherwise,\end{cases}\tag{17}$$

where $\mathcal{L}_{[M]}$ denotes the M-th maximum value of $\{\mathcal{L}^{(i)}\}_{i=1}^{V}$.

Update $\mathbf{F}$: When $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ are fixed, the (9) can be rewritten as:

$$\max_{\mathbf{F}}\mathrm{Tr}(\mathbf{F}^\top\mathbf{G}\mathbf{F}),\quad s.t.\,\mathbf{F}^\top\mathbf{F}=\mathbf{I}.\tag{18}$$

where

$$\mathbf{G}=\sum_{i=1}^{M}\left(\alpha\sum_{j=1}^{P}\mu_i\beta_j\mathbf{H}_j^{(i)}\mathbf{H}_j^{(i)\top}+(1-\alpha)\omega_i\mathbf{S}^{(i)}\right).\tag{19}$$

According to [56], the optimal solution of $\mathbf{F}$ can be obtained by performing eigenvalue decomposition on the matrix $\mathbf{G}$, i.e., $\mathbf{F}$ consists of the eigenvectors corresponding to the first k largest eigenvalues of $\mathbf{G}$.

Update $\boldsymbol{\beta}$: When $\boldsymbol{\gamma}$ and $\mathbf{F}$ are fixed, the (9) is reformulated as:

$$\max_{\boldsymbol{\beta}}\sum_{j=1}^{P}\beta_j\Theta_j,\quad s.t.\,\boldsymbol{\beta}\geq\mathbf{0},\boldsymbol{\beta}^\top\boldsymbol{\beta}=1.\tag{20}$$

where $\Theta_j=\mathrm{Tr}(\mathbf{F}\mathbf{F}^\top\sum_{i=1}^{M}\mu_i\mathbf{H}_j^{(i)}\mathbf{H}_j^{(i)\top})$, and the optimal solution of the above function can be obtained as:

$$\beta_j=\frac{\Theta_j}{\sqrt{\sum_{j=1}^{P}\Theta_j^2}}.\tag{21}$$

In a nutshell, the whole optimization process can be summarized in Algorithm 1.

A. Convergence Analysis

To simplify the expression, we reformulate (9) as follows:

$$\min_{\mathbf{F},\boldsymbol{\gamma},\boldsymbol{\beta}}\boldsymbol{\gamma}\mathcal{L}(\mathbf{F},\boldsymbol{\beta}).\tag{22}$$

Suppose we have obtained $\boldsymbol{\gamma}^{t-1}$, $\mathbf{F}^{t-1}$, and $\boldsymbol{\beta}^{t-1}$ in the $t-1$-th iteration. When we update $\boldsymbol{\gamma}^t$ w.r.t. $\mathbf{F}^{t-1}$, and $\boldsymbol{\beta}^{t-1}$, the following inequality holds:

$$\boldsymbol{\gamma}^t\mathcal{L}(\mathbf{F}^{t-1},\boldsymbol{\beta}^{t-1})\leq\boldsymbol{\gamma}^{t-1}\mathcal{L}(\mathbf{F}^{t-1},\boldsymbol{\beta}^{t-1}).\tag{23}$$

After obtained $\boldsymbol{\gamma}^t$, problem (9) can be expressed as:

$$\begin{aligned}&\min_{\mathbf{F},\boldsymbol{\beta}}\boldsymbol{\gamma}^t\mathcal{L}(\mathbf{F},\boldsymbol{\beta})\\&\Leftrightarrow\min_{\mathbf{F},\boldsymbol{\beta}}\sum_{i=1}^{M}\alpha\mathcal{L}_1^{(i)\frac{1}{2}}(\mathbf{F},\boldsymbol{\beta})+(1-\alpha)\mathcal{L}_2^{(i)\frac{1}{2}}(\mathbf{F})\\&\Leftrightarrow\min_{\substack{\mathbf{F},\boldsymbol{\beta},\\\boldsymbol{\mu},\boldsymbol{\omega}}}\sum_{i=1}^{M}\alpha\mu_i\mathcal{L}_1^{(i)}(\mathbf{F},\boldsymbol{\beta})+(1-\alpha)\omega_p\mathcal{L}_2^{(i)}(\mathbf{F})\end{aligned}\tag{24}$$

and then our goal is proving that alternatively updating $\mathbf{F}$, $\boldsymbol{\mu}$, $\boldsymbol{\omega}$, and $\boldsymbol{\beta}$ can make the value of problem (9) monotonically decrease in each iteration.

For updating $\mathbf{F}$, it is obviously that

$$\begin{aligned}&\min_{\mathbf{F}}\sum_{i=1}^{M}\mathcal{L}_1^{(i)}(\mathbf{F}^t,\boldsymbol{\beta}^{t-1})\leq\sum_{i=1}^{M}\mathcal{L}_1^{(i)}(\mathbf{F}^{t-1},\boldsymbol{\beta}^{t-1})\\&\min_{\mathbf{F}}\sum_{i=1}^{M}\mathcal{L}_2^{(i)}(\mathbf{F}^t)\leq\sum_{i=1}^{M}\mathcal{L}_2^{(i)}(\mathbf{F}^{t-1}).\end{aligned}\tag{25}$$

For updating $\boldsymbol{\beta},\boldsymbol{\mu},\boldsymbol{\omega}$, their optimization problem are all linear convex functions, and closed-form solutions of them are directly obtained.

Algorithm 1: The Algorithm of MPS.

Input: Base kernel matrices $\{\mathbf{K}^{(i)}\}_{i=1}^{V}$, number of selected kernels M and scales P, hyper-parameter α.

1: Construct $\mathbf{H}_j^{(i)}$ and $\mathbf{S}^{(i)}$ via solving (1) and (6), respectively.
2: Initialize $\mathbf{F}$ to zero matrix, $\boldsymbol{\beta} = \frac{1}{\sqrt{P}}$, $t = 1$.
3: **while not converged do**
4: Update $\boldsymbol{\gamma}$ via (17).
5: Update $\mathbf{F}$ via solving (18).
6: Update $\boldsymbol{\beta}$ via (21).
7: $t = t + 1$.
8: **end while**
9: Conduct k-means clustering on the generated consensus partition matrix $\mathbf{F}$.

Output: Clustering results $\mathbf{Y}$.

Combing (23), (24), and (25), the following inequality holds:

$$\boldsymbol{\gamma}^t \mathcal{L}(\mathbf{F}^t, \boldsymbol{\beta}^t) \leq \boldsymbol{\gamma}^{t-1} \mathcal{L}(\mathbf{F}^{t-1}, \boldsymbol{\beta}^{t-1}). \tag{26}$$

Thus, the objective value of Algorithm 1 monotonically decreases in each iteration. Furthermore, it is obvious that the problem (9) has a lower bound, our proposed method can reach convergence.

B. Time Complexity Analysis

In the pre-processing stage, the construction of multi-scale partition matrices through (1) needs a computational complexity of $\mathcal{O}(n^3)$. Regarding the optimization processes, the primary computational burden is associated with solving for $\boldsymbol{\gamma}$, $\mathbf{F}$, and $\boldsymbol{\beta}$. To update $\boldsymbol{\gamma}$, the construction of the values of $\mathcal{L}_1$ and $\mathcal{L}_2$ involves a complexity of $\mathcal{O}(n^2)$. Updating $\mathbf{F}$ has a computational complexity of $\mathcal{O}(n^3)$, and updating β incurs a cost of $\mathcal{O}(P)$, where P denotes the number of scales. Lastly, the k-means clustering step entails a cost of $\mathcal{O}(nk^2)$. Consequently, the overall complexity of the proposed method is $\mathcal{O}(n^3)$.

V. Experiments

A. Datasets

In the experiments, we employ five distinct kernel functions—namely, Gaussian kernel, Polynomial kernel, Linear Kernel, Sigmoid kernel, and InvPloyPlus—on each view of eight diverse multi-view datasets: 3sources,[1] MSRC_v1,[2] BRCA,[3] ORL [57], BBCSport,[4] Caltech101-20,[5] Scene15_3v [58], and EMNIST.[6] These kernel functions are applied to generate multiple kernel datasets, and detailed descriptions of the datasets are provided in Table I, where #Samples, #Kernels, and #Classes indicate the number of samples, kernels, and classes, respectively.

[1]http://mlg.ucd.ie/datasets/3sources.html
[2]https://www.microsoft.com/en-us/research/project/
[3]https://www.cancer.gov/ccg/research/genome-sequencing/tcga
[4]http://mlg.ucd.ie/datasets/segment.html
[5]http://www.vision.caltech.edu/archive.html
[6]https://www.nist.gov/itl/products-and-services/emnist-dataset

TABLE I
Summary of Eight Benchmark Datasets

Dataset	#Samples	#Kernels	#Classes
3sources	169	15	6
MSRC_v1	210	25	7
BRCA	398	20	4
ORL	400	15	40
BBCSport	544	10	5
Caltech101-20	2368	30	20
Scene15_3v	4485	15	15
EMNIST	10000	20	10

B. Compared Methods

To verify the effectiveness of our proposed method, nine multiple kernel clustering methods are selected as the competitors, including:

SB-KKM [42]: The final clustering results are obtained by individually applying the kernel k-means to each base kernel.

A-MKKM [42]: It generates the optimal kernel by directly averaging the base kernels, followed by the application of kernel k-means on the derived one to obtain the clustering results.

MKKM [45]: It represents a conventional k-means clustering approach that incorporates multiple kernels while tuning the kernel weights.

MKKM-MR [59]: It incorporates matrix-induced regularization to optimize the kernel with reduced redundancy.

MVC-LFA [29]: It derives a consensus partition matrix through the alignment with multiple base ones, employing appropriate weights in the process.

SMKKM [60]: Minimizing the kernel alignment criterion is pursued for the acquisition of kernel weights while maximizing it is aimed at deriving the clustering partition matrix.

MKKM-SR [61]: Concurrently conducting multiple kernel k-means and spectral rotation to directly generate the final clustering results.

LSWMKC [62]: A unified affinity graph is formulated in kernel space to capture the latent structures of the original data. Subsequently, clustering benefits from the acquisition of an optimal neighborhood kernel guided by these latent local structures.

EEOMVC [63]: A one-step clustering manner is adopted to generate the final results from the unified partition matrix.

C. Experimental Setup

In the proposed method, the scale parameter P is consistently set to 5 in the conducted experiments, i.e., the range of scale dimension is $[k, 2k, 3k, 4k, 5k]$, resulting in only two parameters requiring determination: the number of selected kernels M and a hyper-parameter α. Due to the inherent challenge of pinpointing optimal values for these parameters empirically, a grid search strategy is employed. Specifically, the tuning of M and α spans the range of 10%-90% of the total number of kernels and $[0.1 : 0.9]$, respectively. For SB-KKM, A-MKKM, MKKM, and SMKKM, which are parameterless methods, we simply downloaded their code and conducted experiments

TABLE II
CLUSTERING RESULTS IN TERMS OF ACC ON EIGHT MULTIPLE KERNEL DATASETS

ACC(%)	3sources	MSRC_v1	BRCA	ORL	BBCSport	Caltech101-20	Scene15_3v	EMNIST
SB-KKM	58.02±0.68	70.71±0.61	59.48±0.14	81.41±2.46	87.32±0.00	41.43±2.14	38.60±0.87	75.07±0.05
A-MKKM	57.22±2.04	69.24±1.99	52.20±0.42	78.63±2.66	80.77±1.03	37.70±1.70	36.00±1.20	63.99±0.08
MKKM	56.98±2.32	62.79±0.86	36.72±0.17	74.27±2.22	80.97±0.76	27.90±0.74	34.94±1.57	59.28±0.23
MKKM-MR	64.73±4.45	84.45±0.60	59.27±0.14	83.48±1.76	91.91±0.00	40.62±2.43	41.32±0.72	73.72±0.02
MVC-LFA	60.41±2.23	77.38±0.39	60.80±0.08	79.79±2.28	90.99±0.00	40.45±1.61	41.16±0.88	69.29±0.09
SMKKM	59.29±1.30	78.40±1.82	57.99±0.48	80.75±2.19	95.40±0.00	36.93±1.50	39.83±1.04	61.45±0.05
MKKM-SR	53.58±4.70	60.95±5.60	51.01±0.90	73.73±3.04	76.53±5.82	37.68±1.85	34.02±2.09	56.83±2.10
LSWMKC	66.95±0.22	83.98±0.28	46.87±0.25	82.50±1.41	**97.98±0.00**	40.54±0.92	37.82±0.65	71.04±0.01
EEOMVC	53.85±0.00	73.81±0.00	58.54±0.00	80.75±0.00	86.95±0.00	43.84±0.00	37.79±0.00	68.46±0.00
MPS	**68.91±0.30**	**86.19±0.00**	**68.59±0.00**	**84.21±1.77**	95.51±0.15	**54.57±1.38**	**48.30±0.08**	**84.23±2.60**

TABLE III
CLUSTERING RESULTS IN TERMS OF NMI ON EIGHT MULTIPLE KERNEL DATASETS

NMI(%)	3sources	MSRC_v1	BRCA	ORL	BBCSport	Caltech101-20	Scene15_3v	EMNIST
SB-KKM	59.69±1.64	59.66±2.24	33.89±0.05	91.86±1.06	74.33±0.00	58.90±0.65	38.00±0.31	67.31±0.06
A-MKKM	64.66±2.17	61.94±2.57	33.29±0.46	89.61±1.30	75.42±1.19	56.90±0.61	33.54±0.55	58.04±0.07
MKKM	63.99±2.59	51.66±1.08	6.18±0.18	87.21±1.15	75.22±0.86	32.27±0.40	32.46±0.68	54.38±0.14
MKKM-MR	65.17±2.08	74.08±1.23	36.99±0.15	92.82±1.14	83.14±0.00	59.41±0.88	39.82±0.31	66.89±0.22
MVC-LFA	64.84±1.80	68.13±0.81	38.18±0.10	90.63±1.38	81.37±0.00	58.67±0.79	38.27±0.26	61.73±0.08
SMKKM	60.44±1.62	68.78±1.31	31.65±0.61	91.02±1.21	**86.31±0.00**	57.44±1.09	36.80±0.28	62.08±0.04
MKKM-SR	52.24±3.96	52.07±3.29	32.78±0.39	86.89±1.67	67.78±6.34	48.51±1.11	29.97±1.23	52.26±1.10
LSWMKC	**68.01±0.57**	75.08±0.26	23.45±0.00	90.15±0.42	93.18±0.00	56.64±0.56	38.58±0.37	72.73±0.02
EEOMVC	53.95±0.00	67.52±0.00	38.68±0.00	91.73±0.00	77.00±0.00	58.12±0.00	34.81±0.00	58.90±0.00
MPS	64.04±5.49	**77.65±2.52**	**45.44±0.00**	**93.33±0.78**	86.04±0.48	**64.10±0.72**	**45.74±0.07**	**81.40±0.01**

TABLE IV
CLUSTERING RESULTS IN TERMS OF ARI ON EIGHT MULTIPLE KERNEL DATASETS

ARI(%)	3sources	MSRC_v1	BRCA	ORL	BBCSport	Caltech101-20	Scene15_3v	EMNIST
SB-KKM	43.42±1.83	51.05±2.55	29.40±0.15	75.70±2.57	76.62±0.00	29.27±1.31	21.63±0.58	60.72±0.07
A-MKKM	45.86±3.48	52.99±3.15	25.90±0.43	71.32±3.33	75.68±0.43	27.42±1.21	18.53±0.87	47.50±0.10
MKKM	44.91±3.76	39.12±0.87	2.75±0.06	65.57±2.96	75.62±0.31	14.04±0.47	19.23±0.91	42.39±0.21
MKKM-MR	54.60±5.13	69.72±1.17	29.59±0.13	78.43±2.96	86.26±0.00	28.98±1.78	23.82±0.48	58.96±0.03
MVC-LFA	50.08±3.22	59.36±0.76	31.54±0.02	73.10±3.37	84.72±0.00	28.69±1.20	23.49±0.25	53.65±0.09
SMKKM	46.50±1.85	59.97±2.03	25.85±0.64	74.32±2.88	88.37±0.00	26.72±1.22	21.66±0.48	49.05±0.08
MKKM-SR	32.27±5.96	40.24±4.43	24.11±0.27	64.23±3.93	63.65±11.04	24.42±1.44	15.98±1.21	39.45±2.19
LSWMKC	54.27±0.19	68.01±0.44	17.49±0.41	74.34±1.13	94.42±0.00	24.49±0.90	21.24±0.22	59.04±0.01
EEOMVC	33.88±0.00	59.00±0.00	31.28±0.00	75.05±0.00	79.80±0.00	30.02±0.00	20.12±0.00	49.02±0.00
MPS	**58.33±1.10**	**72.39±0.00**	**37.69±0.60**	**78.76±2.60**	**88.59±0.40**	**37.28±3.67**	**30.91±0.10**	**74.77±0.01**

accordingly. As for MKKM-MR, MVC-LFA, and MKKM-SR, they involve a single parameter, which we set to range from $[2^{-15} : 2^{15}]$ following the parameterization instructions provided in their respective papers. In the case of LSWMKC, we varied the parameter λ within the range of $[1, 2^{10}]$. For EEOMVC, which has two parameters, λ and β, we set them to $[0.01, 0.05, 0.1, 0.5, 1]$. To assess the clustering performance of all compared methods across various datasets, four metrics are employed in the experiments, including accuracy (ACC), normalized mutual information (NMI), adjusted rand index (ARI), and Fscore. To ensure the robustness of the experimental results, all methods are subjected to 20 repetitions, and all experiments are conducted using Matlab R2022a on an Intel Core i7-13700F CPU with 64 GB RAM.

D. Results

Tables II-V present the clustering results of all compared methods across eight benchmark datasets. In order to clearly present the clustering performance between the different competitors, the best results in terms of each metric are bolded. Based on these results, we can conclude that:

(1) *Refined Kernel Fusion Strategy:* Our proposed method highlights a strategic refinement in kernel fusion compared to SB-KKM and A-MKKM. While SB-KKM assigns equal weighting to each kernel, the combination of all kernels in A-MKKM results in a notable performance decrement, particularly due to the influence of potentially inferior kernels. By adopting a selective approach to exclude such kernels, our

TABLE V
CLUSTERING RESULTS IN TERMS OF FSCORE ON EIGHT MULTIPLE KERNEL DATASETS

Fscore(%)	3sources	MSRC_v1	BRCA	ORL	BBCSport	Caltech101-20	Scene15_3v	EMNIST
SB-KKM	54.50±1.38	58.01±2.17	48.75±0.11	76.27±2.50	82.10±0.00	35.04±1.28	26.98±0.54	64.88±0.07
A-MKKM	56.64±2.80	59.59±2.70	46.19±0.31	72.00±3.25	81.37±0.36	33.19±1.12	24.13±0.82	53.06±0.09
MKKM	55.90±3.03	47.74±0.74	29.86±0.05	66.39±2.89	81.31±0.26	20.82±0.47	24.80±0.84	48.48±0.18
MKKM-MR	64.51±3.90	73.93±1.01	48.97±0.10	78.94±2.88	89.54±0.00	34.64±1.69	29.04±0.44	63.32±0.02
MVC-LFA	60.08±2.64	65.05±0.66	50.15±0.02	73.74±3.29	88.35±0.00	34.42±1.11	28.76±0.24	58.56±0.08
SMKKM	57.31±1.61	65.54±1.75	45.99±0.46	74.92±2.81	91.15±0.00	32.81±1.16	27.05±0.43	54.65±0.07
MKKM-SR	47.43±4.30	48.62±3.82	45.21±0.16	65.09±3.83	73.56±7.24	30.51±1.40	21.85±1.14	45.90±1.93
LSWMKC	63.31±0.14	72.58±0.37	40.22±0.29	74.94±1.10	95.74±0.00	30.62±1.48	26.65±0.22	63.60±0.01
EEOMVC	50.37±0.00	64.70±0.00	50.14±0.00	75.65±0.00	84.84±0.00	35.77±0.00	25.70±0.00	54.15±0.00
MPS	**67.87±0.75**	**76.25±0.00**	**55.40±0.00**	**79.26±2.53**	**91.26±0.30**	**44.89±1.64**	**35.72±0.09**	**77.31±0.01**

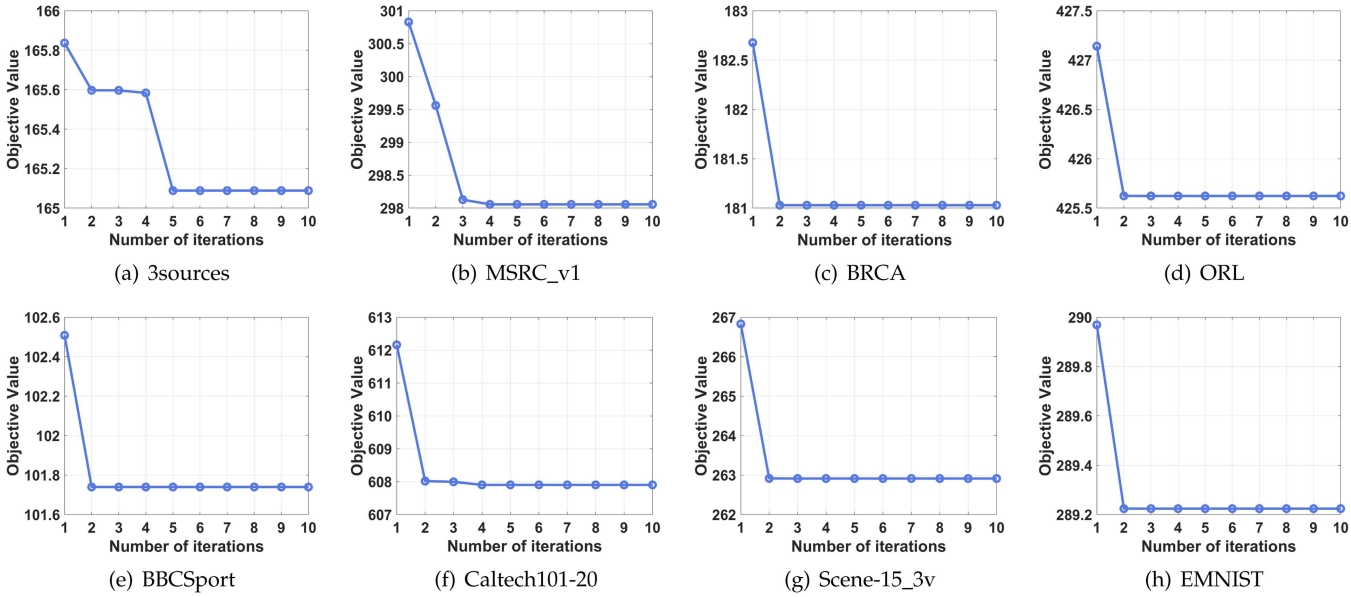


Fig. 3. Objective function values at each iteration of the proposed method on eight datasets.

method demonstrates superior performance across the majority of datasets examined.

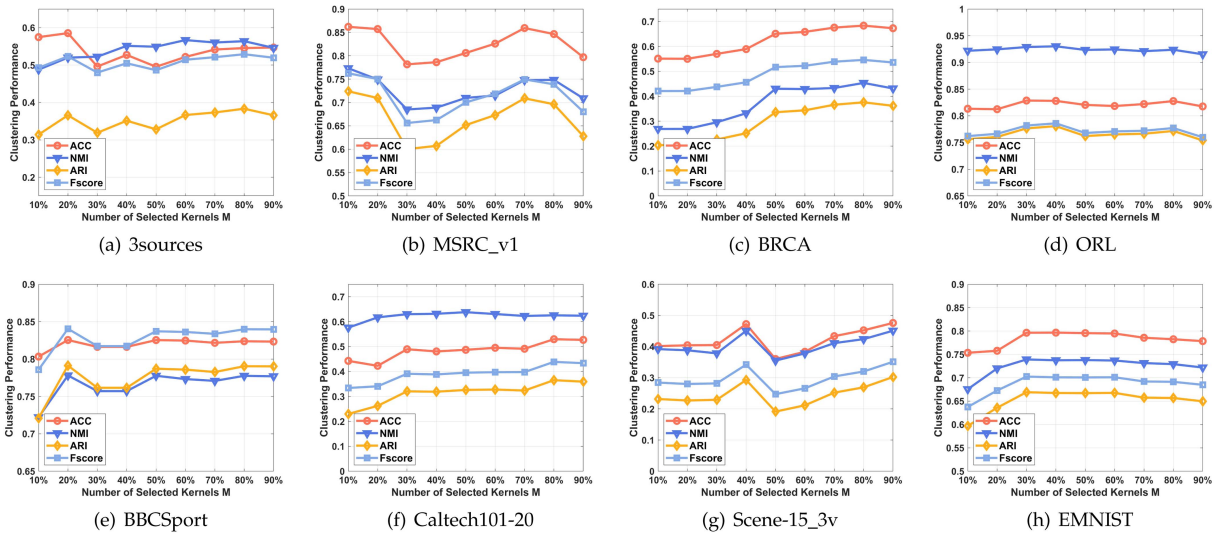
(2) *Enhanced Partition Generation:* In contrast to LSWMKC, which relies on consensus affinity graphs derived from the kernel space, our approach innovatively incorporates multi-scale partitions alongside a similarity graph. This integration markedly enhances the effectiveness of the resultant optimal kernel. Empirical results from the comparative tables consistently demonstrate our method's outperformance of LSWMKC across a majority of evaluated datasets, emphasizing the significance and innovation of our strategy for generating multi-scale partitions.

(3) *Emphasis on Structural Information Learning:* Unlike prevailing methods within the MKC paradigm, our approach consistently delivers superior performance across diverse datasets. This commendable performance stems from our method's strategic emphasis on harnessing inherent structural information among samples. While comparative methods primarily focus on achieving a consensus partition using all available kernels, our approach astutely leverages structural interconnectivity, resulting in notably enhanced performance. This nuanced distinction not only underscores the efficacy of our method but also highlights potential limitations in the singular focus on kernel consensus within traditional MKC frameworks.

E. Convergence and Parameter Sensitivity Analysis

As mentioned previously, our proposed method demonstrates rapid convergence within a few iterations. In this section, we substantiate this claim through experimentation, and the corresponding results are presented in Fig. 3. The depicted figure illustrates a sharp decrease in the objective function's value, attaining convergence swiftly. This unequivocally validates the efficacy of our designed solution algorithms.

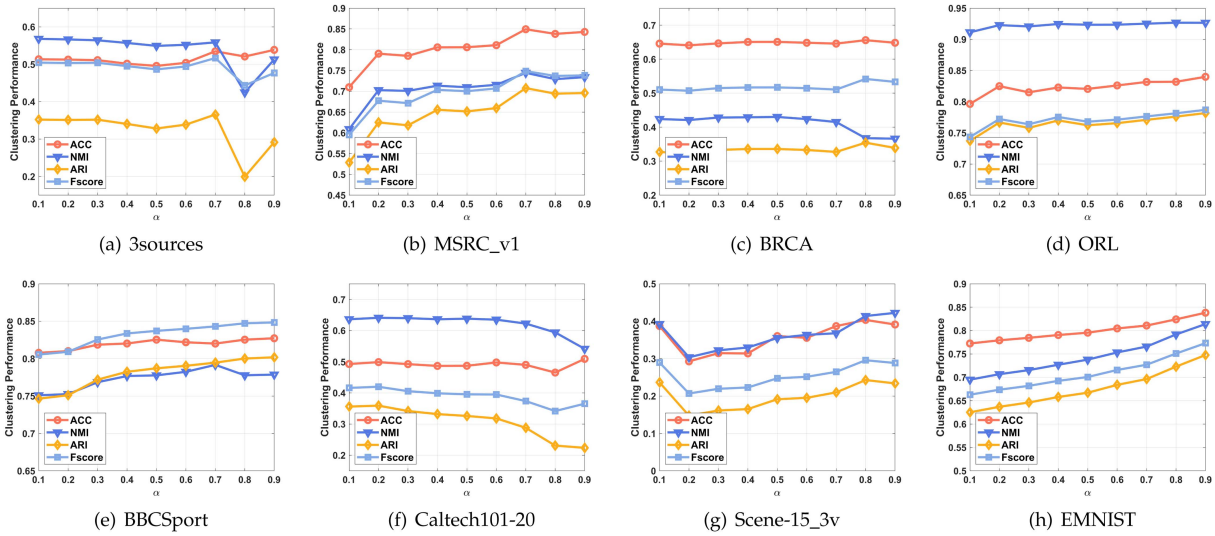
In our experimental evaluation, our method involves two key parameters: the number of selected kernels, denoted as M, and the hyper-parameter α. To assess the performance of MPS across



(a) 3sources (b) MSRC_v1 (c) BRCA (d) ORL

(e) BBCSport (f) Caltech101-20 (g) Scene-15_3v (h) EMNIST

Fig. 4. The parameter sensitivity analysis of MPS on the parameter M when α is fixed to 0.5.



(a) 3sources (b) MSRC_v1 (c) BRCA (d) ORL

(e) BBCSport (f) Caltech101-20 (g) Scene-15_3v (h) EMNIST

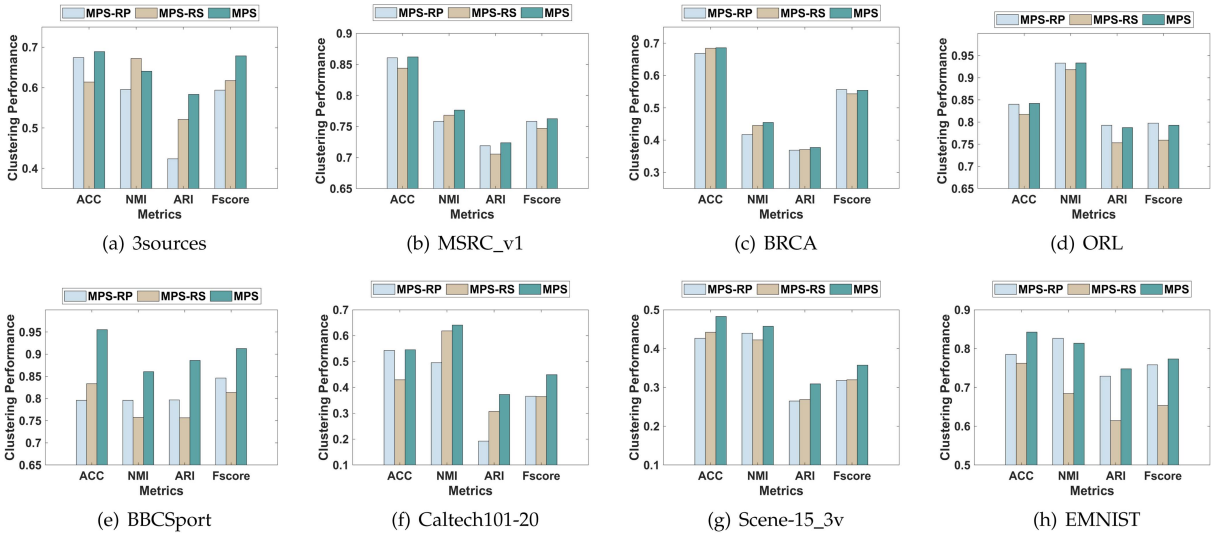
Fig. 5. The parameter sensitivity analysis of MPS on the parameter α when M is fixed to 50% of the total number of kernels.

different parameter values, we conducted experiments by fixing one parameter while optimizing the other. The results of these experiments are presented in Figs. 4 and 5.

As mentioned in the previous discussions, the parameter α is employed to regulate the balance between the pairwise relationships among samples and cluster structures. Consequently, varying its values leads to an adjustment of the weight distribution that leverages information from both aspects. However, across different multiple kernel datasets, the nature of pairwise relationships among samples or cluster structures manifests variability. For instance, certain datasets may exhibit a prevalence of pairwise relationships among samples, while others may emphasize information pertaining to cluster structures. Consequently, altering the weights between these aspects may introduce instability in the final clustering performance, underscoring the sensitivity of the parameters. Although kernel matrices inherently possess a degree of information redundancy, they retain specific information crucial for enhancing clustering performance. Hence, opting for fewer kernels results in diminished clustering performance due to underutilization of available information. Conversely, selecting an excessive number of kernels introduces redundant information, thereby yielding diminishing returns in terms of information gain. This phenomenon elucidates the fluctuating trends observed in Fig. 4 when varying the number of kernels.

F. Ablation Study

In our proposed method, we seamlessly integrate multi-scale partitions and a similarity graph within a unified framework to facilitate the optimization of the consensus partition. To



(a) 3sources (b) MSRC_v1 (c) BRCA (d) ORL

(e) BBCSport (f) Caltech101-20 (g) Scene-15_3v (h) EMNIST

Fig. 6. The ablation experimental results of MPS on eight multiple kernel datasets.

TABLE VI
THE OPTIMAL PARTITION RESULTS ACROSS ALL SCALES AND THE RESULTS OBTAINED USING ALL KERNELS OF THE PROPOSED METHOD

Metrics	Methods	3sources	MSRC_v1	BRCA	ORL	BBCSport	Caltech101-20	Scene15_3v	EMNIST
ACC	Best Partition	60.83±4.41	72.00±6.19	59.70±5.37	81.41±2.46	87.32±0.00	42.94±2.36	40.35±0.95	81.02±1.56
	MPS(w/o KS)	64.94±2.50	85.24±0.00	66.92±0.17	83.29±1.59	82.54±0.00	49.44±3.91	**48.40±0.13**	81.61±0.09
	MPS	**68.91±0.30**	**86.19±0.00**	**68.59±0.00**	**84.21±1.77**	**95.51±0.15**	**54.57±1.38**	48.30±0.08	**84.23±2.60**
NMI	Best Partition	59.73±4.75	64.08±5.22	34.85±0.86	91.86±1.06	76.64±5.60	62.72±1.12	40.13±0.97	75.02±1.21
	MPS(w/o KS)	62.14±1.77	75.40±0.00	44.90±0.29	91.67±0.58	78.14±0.08	63.10±0.86	**46.65±0.14**	77.46±0.06
	MPS	**64.04±5.49**	**77.65±2.52**	**45.44±0.00**	**93.33±0.78**	**86.04±0.48**	**64.10±0.72**	45.74±0.07	**81.40±0.01**
ARI	Best Partition	43.42±1.83	53.76±6.98	29.40±0.15	75.70±2.57	76.62±0.00	31.50±3.17	22.57±0.75	68.33±2.17
	MPS(w/o KS)	51.82±6.23	71.13±0.00	36.48±0.22	76.80±1.92	79.93±0.05	**39.83±1.16**	30.73±0.17	69.47±0.09
	MPS	**58.33±1.10**	**72.39±0.00**	**37.69±0.60**	**78.76±2.60**	**88.59±0.40**	37.28±3.67	**30.91±0.10**	**74.77±0.01**
Fscore	Best Partition	54.60±5.38	60.35±5.94	50.15±4.24	76.27±2.50	82.10±0.00	37.64±3.10	28.12±0.69	71.56±1.93
	MPS(w/o KS)	63.38±4.67	75.16±0.00	54.29±0.04	77.35±1.87	84.65±0.03	**46.09±1.00**	**35.73±0.15**	72.58±0.08
	MPS	**67.87±0.75**	**76.25±0.00**	**55.40±0.00**	**79.26±2.53**	**91.26±0.30**	44.89±1.64	35.72±0.09	**77.31±0.01**

rigorously assess their effectiveness, we conduct ablation experiments in this section. Specifically, we systematically exclude each component individually, utilizing the remaining elements to generate the consensus partition in the experiments. For clarity, the removal of the multi-scale partitions fusion component is denoted as MPS-RP, while the removal of the similarity graph component is denoted as MPS-RS. The detailed experimental results are presented in Fig. 6. According to the results in terms of four evaluation metrics, we can find that the proposed method consistently outperforms the two variant methods across all datasets through the synergistic contribution of these two pivotal components. This highlights the beneficial impact of both components on the overall clustering performance.

G. Multi-Scale Partition and Kernel Selection Study

In the proposed method, we adopt multi-scale partition generation and kernel selection to improve the final clustering performance. To further verify their effectiveness, the corresponding experiments are conducted and the results are reported in Table VI. In Table VI, the Best Partition denotes the optimal kernel results at all scales, while MPS (w/o KS) represents the results of the proposed method without kernel selection. As seen in these results, it can be observed that the introduction of multi-scale partition integration and kernel selection strategy is indeed effective in enhancing clustering performance. Therefore, we can conclude that the utilization of multi-scale partition generation and kernel selection are benefits for multiple kernel clustering.

VI. CONCLUSION

This paper introduces an innovative MKC method, characterized by an adaptive multi-scale partition selection mechanism. Specifically, our proposed approach aims to bridge the information loss gap observed during partition generation by constructing multi-scale partitions. These partitions are designed to

encode the complementary information inherent in each kernel, thus offering a more comprehensive view of the data landscape. Additionally, to mitigate the impact of kernels that contribute negatively to the final clustering results, we employ a dynamic adaptive kernel selection strategy during the partition fusion phase. This strategy ensures the preservation of kernels that are deemed to be of relatively higher quality, thereby enhancing the robustness of the clustering process. To further refine the clustering framework, we introduce a kernel-based similarity graph. This graph is instrumental in distilling the structural relationships among samples, thereby furnishing a more informed basis for consensus partition generation.

For the kernel selection problem, one of the most critical questions is how to determine the optimal number of selected kernels. In this paper, the proposed method regards it as a parameter and then determines it by employing a grid search. This approach is shallow. Thus, we will focus on how to make the proposed method determine the optimal number of kernels automatically in the future.

REFERENCES

[1] C. Biemann, "Chinese whispers-an efficient graph clustering algorithm and its application to natural language processing problems," in *Proc. TextGraphs: 1st Workshop Graph Based Methods Natural Lang. Process.*, 2006, pp. 73–80.

[2] A.-C. Ngonga Ngomo and F. Schumacher, "Borderflow: A local graph clustering algorithm for natural language processing," in *Proc. Int. Conf. Intell. Text Process. Comput. Linguistics*, 2009, pp. 547–558.

[3] Y. Hu, H. Mao, and G. McKenzie, "A natural language processing and geospatial clustering framework for harvesting local place names from geotagged housing advertisements," *Int. J. Geographical Inf. Sci.*, vol. 33, no. 4, pp. 714–738, 2019.

[4] R. Guan, H. Zhang, Y. Liang, F. Giunchiglia, L. Huang, and X. Feng, "Deep feature-based text clustering and its explanation," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 8, pp. 3669–3680, Aug. 2022.

[5] K. Liang et al., "Knowledge graph contrastive learning based on relation-symmetrical structure," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 1, pp. 226–238, Jan. 2024.

[6] R. T. Ng and J. Han, "CLARANS: A method for clustering objects for spatial data mining," *IEEE Trans. Knowl. Data Eng.*, vol. 14, no. 5, pp. 1003–1016, Sep./Oct. 2002.

[7] F. Yang, H. Fan, P. Chu, E. Blasch, and H. Ling, "Clustered object detection in aerial images," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 8311–8320.

[8] G. Stockman, S. Kopstein, and S. Benett, "Matching images to models for registration and object detection via clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-4, no. 3, pp. 229–241, May 1982.

[9] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han, "Outlier detection for temporal data: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 9, pp. 2250–2267, Sep. 2014.

[10] A. MartÍnez-UsÓMartinez-Uso, F. Pla, J. M. Sotoca, and P. GarcÍa-Sevilla, "Clustering-based hyperspectral band selection using information measures," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 12, pp. 4158–4171, Dec. 2007.

[11] C. Tang et al., "Spatial and spectral structure preserved self-representation for unsupervised hyperspectral band selection," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5531413.

[12] J. Wang et al., "Hyperspectral band selection via region-aware latent features fusion based clustering," *Inf. Fusion*, vol. 79, pp. 162–173, 2022.

[13] R. Guan et al., "Contrastive multiview subspace clustering of hyperspectral images based on graph convolutional networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5510514.

[14] D. Hu, K. Liang, Z. Dong, J. Wang, Y. Zhao, and K. He, "Effective multi-modal clustering method via skip aggregation network for parallel scRNA-seq and scATAC-seq data," *Brief. Bioinf.*, vol. 25, no. 2, 2024, Art. no. bbae102.

[15] D. Hu, K. Liang, S. Zhou, W. Tu, M. Liu, and X. Liu, "SCDFC: A deep fusion clustering method for single-cell RNA-SEQ data," *Brief. Bioinf.*, vol. 24, no. 4, 2023, Art. no. bbad 216.

[16] H.-P. Kriegel, P. Kröger, and A. Zimek, "Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering," *ACM Trans. Knowl. Discov. Data*, vol. 3, no. 1, pp. 1–58, 2009.

[17] K. Zhan, C. Niu, C. Chen, F. Nie, C. Zhang, and Y. Yang, "Graph structure fusion for multiview clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 10, pp. 1984–1993, Oct. 2019.

[18] Z. Lin, Z. Kang, L. Zhang, and L. Tian, "Multi-view attributed graph clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 2, pp. 1872–1880, Feb. 2023.

[19] Z. Li, C. Tang, X. Liu, X. Zheng, W. Zhang, and E. Zhu, "Consensus graph learning for multi-view clustering," *IEEE Trans. Multimedia*, vol. 24, pp. 2461–2472, 2022.

[20] J. Wen et al., "Adaptive graph completion based incomplete multi-view clustering," *IEEE Trans. Multimedia*, vol. 23, pp. 2493–2504, 2021.

[21] C. Tang, Z. Li, J. Wang, X. Liu, W. Zhang, and E. Zhu, "Unified one-step multi-view spectral clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 6, pp. 6449–6460, Jun. 2023.

[22] Z. Ren, S. X. Yang, Q. Sun, and T. Wang, "Consensus affinity graph learning for multiple kernel clustering," *IEEE Trans. Cybern.*, vol. 51, no. 6, pp. 3273–3284, Jun. 2021.

[23] X. Xie, X. Guo, G. Liu, and J. Wang, "Implicit block diagonal low-rank representation," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 477–489, Jan. 2018.

[24] X. Liu et al., "Optimal neighborhood kernel clustering with multiple kernels," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 2266–2272.

[25] S. Wang, X. Liu, L. Liu, S. Zhou, and E. Zhu, "Late fusion multiple kernel clustering with proxy graph refinement," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 8, pp. 4359–4370, Aug. 2023.

[26] J. Wang et al., "Fast approximated multiple kernel k-means," *IEEE Trans. Knowl. Data Eng.*, 2023, early access, Dec. 13, 2023, doi: 10.1109/TKDE.2023.3340743.

[27] C.-D. Wang, M.-S. Chen, L. Huang, J.-H. Lai, and S. Y. Philip, "Smoothness regularized multiview subspace clustering with kernel learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 11, pp. 5047–5060, Nov. 2021.

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

[29] S. Wang et al., "Multi-view clustering via late fusion alignment maximization," in *Proc. Int. Joint Conf. Artif. Intell.*, 2019, pp. 3778–3784.

[30] X. Liu et al., "Late fusion incomplete multi-view clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 10, pp. 2410–2423, Oct. 2019.

[31] X. Liu et al., "Multiple kernel K K-means with incomplete kernels," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 5, pp. 1191–1204, May 2020.

[32] Y. Zhang, X. Liu, S. Wang, J. Liu, S. Dai, and E. Zhu, "One-stage incomplete multi-view clustering via late fusion," in *Proc. 29th ACM Int. Conf. Multimedia*, 2021, pp. 2717–2725.

[33] S. Wang et al., "Align then fusion: Generalized large-scale multi-view clustering with anchor matching correspondences," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 5882–5895.

[34] T. Wang, L. Zhang, and W. Hu, "Bridging deep and multiple kernel learning: A review," *Inf. Fusion*, vol. 67, pp. 3–13, 2021.

[35] C. Wu, Z. Khan, S. Ioannidis, and J. G. Dy, "Deep kernel learning for clustering," in *Proc. SIAM Int. Conf. Data Mining*, 2020, pp. 640–648.

[36] H. Song, J. J. Thiagarajan, P. Sattigeri, and A. Spanias, "Optimizing kernel machines using deep learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 11, pp. 5528–5540, Nov. 2018.

[37] Y. Wang, X. Liu, Y. Dou, and R. Li, "Approximate large-scale multiple kernel K-means using deep neural network," in *Proc. Int. Joint Conf. Artif. Intell.*, 2017, pp. 3006–3012.

[38] P. Ji, T. Zhang, H. Li, M. Salzmann, and I. Reid, "Deep subspace clustering networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 23–32.

[39] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 478–487.

[40] J. Liu, X. Liu, S. Wang, S. Zhou, and Y. Yang, "Hierarchical multiple kernel clustering," in *Proc. AAAI Conf. Artif. Intell.*, vol. 2021, pp. 8671–8679.

[41] Z. Li et al., "Mutual structure learning for multiple kernel clustering," *Inf. Sci.*, vol. 647, 2023, Art. no. 119445.

[42] I. S. Dhillon, Y. Guan, and B. Kulis, "Kernel K-means: Spectral clustering and normalized cuts," in *Proc. 10th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2004, pp. 551–556.

[43] M. Yin, Y. Guo, J. Gao, Z. He, and S. Xie, "Kernel sparse subspace clustering on symmetric positive definite manifolds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 5157–5164.

[44] R. Zhang and A. I. Rudnicky, "A large scale clustering scheme for kernel k-means," in *Proc. Int. Conf. Pattern Recognit.*, 2002, pp. 289–292.

[45] H.-C. Huang, Y.-Y. Chuang, and C.-S. Chen, "Multiple kernel fuzzy clustering," *IEEE Trans. Fuzzy Syst.*, vol. 20, no. 1, pp. 120–134, Feb. 2012.
[46] P. Zhou, L. Du, L. Shi, H. Wang, and Y.-D. Shen, "Recovery of corrupted multiple kernels for clustering," in *Proc. 24th Int. Joint Conf. Artif. Intell.*, 2015, pp. 4105–4111.
[47] S. Yu et al., "Optimized data fusion for kernel k-means clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 5, pp. 1031–1039, May 2012.
[48] M. Gönen and A. A. Margolin, "Localized data fusion for kernel K-means clustering with application to cancer biology," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 1305–1313.
[49] F. Nie, X. Wang, and H. Huang, "Clustering and projected clustering with adaptive neighbors," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2014, pp. 977–986.
[50] T. Liu, C. K. L. Lekamalage, G.-B. Huang, and Z. Lin, "An adaptive graph learning method based on dual data representations for clustering," *Pattern Recognit.*, vol. 77, pp. 126–139, 2018.
[51] D. Wu, W. Chang, J. Lu, F. Nie, R. Wang, and X. Li, "Adaptive-order proximity learning for graph-based clustering," *Pattern Recognit.*, vol. 126, 2022, Art. no. 108550.
[52] X. Li, H. Zhang, and R. Zhang, "Adaptive graph auto-encoder for general data clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 12, pp. 9725–9732, Dec. 2022.
[53] X. Li, M. Chen, and Q. Wang, "Adaptive consistency propagation method for graph clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 4, pp. 797–802, Apr. 2020.
[54] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, Jul. 1998.
[55] F. Nie, S. Shi, J. Li, and X. Li, "Implicit weight learning for multi-view clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 8, pp. 4223–4236, Aug. 2023.
[56] J. Wang et al., "Region-aware hierarchical latent feature representation learning-guided clustering for hyperspectral band selection," *IEEE Trans. Cybern.*, vol. 53, no. 8, pp. 5250–5263, Aug. 2023.
[57] P. Zhu, B. Hui, C. Zhang, D. Du, L. Wen, and Q. Hu, "Multi-view deep subspace clustering networks," *IEEE Trans. Cybern.*, pp. 1–14, 2024.
[58] L. Fei-Fei and P. Perona, "A Bayesian hierarchical model for learning natural scene categories," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2005, pp. 524–531.
[59] X. Liu, Y. Dou, J. Yin, L. Wang, and E. Zhu, "Multiple kernel k-means clustering with matrix-induced regularization," in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 1888–1894.
[60] X. Liu, "SimpleMKKM: Simple multiple kernel k-means," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 5174–5186, Apr. 2023.
[61] J. Lu, Y. Lu, R. Wang, F. Nie, and X. Li, "Multiple kernel k-means clustering with simultaneous spectral rotation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2022, pp. 4143–4147.
[62] L. Li et al., "Local sample-weighted multiple kernel clustering with consensus discriminative graph," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 2, pp. 1721–1734, Feb. 2024.
[63] J. Wang, C. Tang, Z. Wan, W. Zhang, K. Sun, and A. Y. Zomaya, "Efficient and effective one-step multiview clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, 2023, early access, Mar. 14, 2023, doi: 10.1109/TNNLS.2023.3253246.

Jun Wang received the MS degree from the China University of Geosciences, Wuhan, China, in 2023. He is currently working toward the PhD degree with the School of Computer Science, National University of Defense Technology. His research focuses on multiview learning.

Zhenglai Li received the BE and MS degrees in 2018 and 2021, respectively, from the China University of Geosciences, Wuhan, China, where he is currently working toward the PhD degree. His research focuses on multiview learning.

Chang Tang (Senior Member, IEEE) received the PhD degree from Tianjin University, Tianjin, China, in 2016. He joined the AMRL Lab of the University of Wollongong between 2014 and 2015. He is currently a full professor with the School of Computer Science, China University of Geosciences, Wuhan, China. He has authored or coauthored more than 50 peer-reviewed papers, including those in highly regarded journals and conferences, such as *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *IEEE Transactions on Multimedia*, *IEEE Transactions on Knowledge and Data Engineering*, *IEEE Transactions on Human-Machine Systems*, ICCV, CVPR, IJCAI, AAAI, and ACM MM. His research interests include machine learning and computer vision. He is an associate editor for *BioMed Research International*, *BMC Bioinformatics*, young editor of *CAAI Transactions on Intelligence Technology* and *Computer Engineering*. He regularly serves on the Technical Program Committees or as the area chair of some top conferences, such as NIPS, ICML, CVPR, ICCV, ECCV, IJCAI, ICME, and AAAI.

Suyuan Liu is currently working toward the PhD degree with the National University of Defense Technology, Changsha, China. He has authored or coauthored several papers and served as a Program Committee (PC) member or a reviewer for top conferences, such as *IEEE Transactions on Knowledge and Data Engineering*, *IEEE Transactions on Neural Networks and Learning Systems*, *IEEE Transactions on Circuits and Systems for Video Technology*, NeurIPS, AAAI, and ACM MM. His research interests include multiple-view learning and scalable clustering.

Xinhang Wan received the BE degree in computer science and technology from Northeastern University, Shenyang, China, in 2021. He is currently working toward the master's degree with the National University of Defense Technology, Changsha, China. He has authored or coauthored papers in journals and conferences, such as *IEEE Transactions on Neural Networks and Learning Systems*, ACMMM, and AAAI. His research interests include multiview learning, continual clustering, and active learning.

Xinwang Liu (Senior Member, IEEE) received the PhD degree from the National University of Defense Technology (NUDT), Changsha, China. He is currently a full professor with the School of Computer, NUDT. Dr. Liu has authored or coauthored more than 60 peer-reviewed papers, including those in highly regarded journals and conferences, such as *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *IEEE Transactions on Knowledge and Data Engineering*, *IEEE Transactions on Image Processing*, *IEEE Transactions on Neural Networks and Learning Systems*, *IEEE Transactions on Multimedia*, *IEEE Transactions on Information Forensics and Security*, NeurIPS, ICCV, CVPR, AAAI, and IJCAI. His research interests include kernel learning and unsupervised feature learning. More information can be found at xinwangliu.github.io.