Fast Approximated Multiple Kernel K-means

Jun Wang, Chang Tang, *Senior Member, IEEE,* Xiao Zheng, Xinwang Liu, *Senior Member, IEEE,* Wei Zhang, *Member, IEEE,* En Zhu, Xinzhong Zhu

Abstract—Multiple Kernel Clustering (MKC) has emerged as a prominent research domain in recent decades due to its capacity to exploit diverse information from multiple views by learning an optimal kernel. Despite the successes achieved by various MKC methods, a significant challenge lies in the computational complexity associated with generating a consensus partition from the optimal kernel matrix, typically of size $n \times n$, where n represents the number of samples. This computational bottleneck restricts the practical applicability of these methods when confronted with large-scale datasets. Furthermore, certain existing MKC algorithms derive the consensus partition matrix by fusing all base partitions. However, this fusion process may inadvertently overlook critical information embedded in individual base kernels, potentially leading to inferior clustering performance. In light of these challenges, we introduce an innovative and efficient multiple kernel k-means approach, denoted as FAMKKM. Notably, FAMKKM incorporates two approximated partition matrices instead of the original individual partition matric for each base kernel. This strategic substitution significantly reduces computational complexity. Additionally, FAMKKM leverages the original kernel information to guide the fusion of all base partitions, thereby enhancing the quality of the resulting consensus partition matrix. Finally, we substantiate the efficacy and efficiency of the proposed FAMKKM through extensive experiments conducted on six benchmark datasets. Our results demonstrate its superiority over state-of-the-art methods. The demo code of this work is publicly available at https://github.com/WangJun2023/FAMKKM.

Index Terms—Multi-view clustering, partition learning, multiple kernel k-means, data fusion.

1 Introduction

Clustering is a fundamental algorithm for assigning each object to its corresponding class, widely used in machine learning and computer vision communities [1]–[7]. With the increasing availability of data from diverse sources, object data can be represented by heterogeneous features, making it necessary to develop advanced multi-view clustering methods to handle this kind of datasets effectively [8]–[15]. Although these methods have shown potential for multi-view data clustering, capturing the intrinsic critical information of large-scale datasets in the era of big data remains a challenging problem.

K-means based clustering has been widely used in clustering analysis due to its simplicity and mathematical tractability. Over the past decades, many variants of k-means clustering methods have been proposed in the literature, such as [16]–[22]. In order to handle complex multi-view datasets, multiple kernel k-means based algorithms (MKKM) have been proposed. For example, Li et al. [23] developed a local kernel alignment method to improve the clustering performance by utilizing the variation among samples. Since multiple kernel clustering aims to utilize a group of pre-defined kernel matrices to explore the multi-view data structure, the selected kernels may become redundant as the number of views increases. To capture the correlations among multiple kernel matrices, Liu et al. [24] integrated a matrix-induced regularization term into the proposed method. In most existing MKKM methods, a common assumption is taken that the unified optimal kernel matrix is generated by using a linear combination of all base kernel matrices [10], [25], [26]. Although this assumption reduces the computational complexity, the representation capability of the unified kernel may be limited. To address this issue, an optimal neighborhood kernel clustering method is proposed in [27] to improve the representability of the unified kernel matrix. Moreover, several MKKM methods have been developed to handle incomplete views, where the rows or columns of base kernels are absent in some scenarios, e.g., [28]–[30].

In the multiple kernel k-means clustering algorithms mentioned above, the objective functions differ in various aspects, but they all need to generate a consensus clustering partition matrix. Based on the stage of clustering information exploitation, existing MKKM methods can be broadly categorized into two categories: early-fusion [31]–[35] and late-fusion [28], [36]–[39]. The first category jointly optimizes the kernel coefficient and the consensus clustering partition matrix based on all kernel matrices, resulting in a better unified partition matrix with the guidance of original kernel information. However, this approach increases computational complexity significantly due to the eigen-decomposition of the kernel matrix with the size of $n \times n$. The second category generates all base partitions independently from each kernel and then fuses them to obtain the consensus partition matrix. This approach derives the consensus partition matrix at the partition level rather

The work was partly supported by the National Natural Science Foundation of China (NO. 62076228, 62325604, and 62376252), and partly supported by Natural Natural Science Foundation of Shandong Province (NO. ZR2021LZH001). (Corresponding author: Chang Tang).

J. Wang, X. Zheng, X. Liu and E, Zhu are with the School of Computer, National University of Defense Technology, Changsha 410073, China (E-mail: wang_jun, zhengxiao, xinwangliu, enzhu@nudt.edu.cn)

C. Tang is with the School of Computer Science, China University of Geosciences, Wuhan 430074, China. He is also with the State Key Laboratory of Integrated Services Networks (Xidian University), Xi'an, 710071, China. (E-mail: tangchang@cug.edu.cn).

W. Zhang is with Shandong Provincial Key Laboratory of Computer Networks, Shandong Computer Science Center (National Supercomputing Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Jinan 250000, China (E-mail: wzhang@qlu.edu.cn).

X. Zhu is with the School of Computer Science and Technology (School of Artificial Intelligence), Zhejiang Normal University, Jinhua 321004, China. (E-mail: zxz@zjnu.edu.cn).

than the kernel level, significantly alleviating computational complexity by introducing a group of smaller base partition matrices. However, the final clustering performance highly depends on the quality of each base partition without the interaction of original kernel information.

Considering that the generation of base partitions from each corresponding kernel matrix results in the loss of complementary information, we propose a joint framework for fusing all base partitions and kernel matrices. This aims to produce a more representative and unified partition matrix, which can enhance the final clustering performance. However, the current challenge lies in obtaining the base partitions from the original kernel matrices with reasonable computational overhead. The typical approach for generating base partitions is through eigen-decomposition, which requires $\mathcal{O}(n^3)$ computational complexity and is challenging to apply to large-scale multi-view datasets. To overcome this, we introduce two approximated partition matrices in the partition generation step and propose a simple and effective multiple kernel k-means clustering method. Furthermore, we theoretically establish the conceptual equivalence between approximated partition generation and existing partition generation. Our proposed method can achieve superior clustering performance with less computational overhead than other state-of-the-art MKKM, as demonstrated by subsequent experimental results.

In summary, the main contributions of this paper include:

- We develop a unified framework that combines partition fusion and kernel fusion, allowing the two steps to complement each other. This approach enables us to generate a more effective partition matrix that incorporates valuable information from the original kernel matrices.
- We propose the use of two approximated partition matrices to mitigate the computational complexity of our proposed method from $\mathcal{O}(n^2)$ or $\mathcal{O}(n^3)$ to $\mathcal{O}(n)$. We present theoretical proof that optimizing the two approximated partitions is conceptually equivalent to optimizing the original partition derived from each kernel independently.
- An alternating iterative optimization algorithm is developed to solve the formulated model, and extensive experimental results demonstrate the efficiency and effectiveness of the proposed method on six benchmark datasets.

The structure of this paper is as follows. Section II provides a brief review of k-means clustering and its variant, multiple kernel k-means clustering. The proposed method and its corresponding optimization solutions are presented in Section III and Section IV, respectively. To demonstrate the effectiveness of the proposed method, Section V describes extensive experiments. Finally, concluding remarks are provided in Section VI.

2 Related Work

2.1 Kernel K-means Clustering

Given a single-view data matrix $\mathbf{X} = [x_1, ..., x_n] \in \mathbb{R}^{n \times d}$, characterized by n samples and d features. We assume that it can be segmented into k clusters denoted as $\mathbf{X} = [X_1, ..., X_k]$, where k represents the number of clusters, and $X_i = [x_1^i, ..., x_{s_i}^i]$ signifies the i-th cluster comprising s_i samples. In this context, the objective function for the k-means clustering algorithm can be formally expressed as follows [40]:

$$\min \sum_{i=1}^{k} \sum_{j=1}^{s_i} \left\| x_j^i - c_i \right\|^2. \tag{1}$$

where $c_i = \sum_{j=1}^{s_i} x_j^i / s_i$ represents the centroid of the i-th cluster. Let us introduce the vector $e_i = [1, ..., 1]^\top \in \mathbb{R}^{s_i}$, enabling us to derive $c_i = \frac{X_i e_i}{s_i}$. Subsequently, we formulate the following expression:

$$\min \sum_{j=1}^{s_i} \left\| x_j^i - c_i \right\|^2 = \min ||X_i(I_{s_i} - e_i e_i^\top / s_i)||_F^2. \tag{2}$$

Due to $(I_{s_i} - e_i e_i^\top / s_i)^2 = (I_{s_i} - e_i e_i^\top / s_i)$, the following equations hold:

$$\begin{aligned} &\min ||X_i(I_{s_i} - e_i e_i^\top / s_i)||_F^2 \\ &= \min \mathrm{Tr}(X_i^\top (I_{s_i} - e_i e_i^\top / s_i) X_i) \\ &= \min \mathrm{Tr}(X_i^\top X_i) - \mathrm{Tr}(X_i^\top (e_i e_i^\top / s_i) X_i). \end{aligned} \tag{3}$$

By introducing an orthogonal matrix $\mathbf{H}$ that satisfies:

$$\mathbf{H} = \begin{bmatrix} \frac{e_1}{\sqrt{s_1}} & & & \\ & \frac{e_2}{\sqrt{s_2}} & & \\ & & \cdots & \\ & & & \frac{e_k}{\sqrt{s_k}} \end{bmatrix}, \tag{4}$$

the Eq. (1) can be reformulated as follows:

$$\min_{\mathbf{H}} \mathrm{Tr}(\mathbf{X}^\top \mathbf{X}) - \mathrm{Tr}(\mathbf{X}^\top \mathbf{H} \mathbf{H}^\top \mathbf{X}) \quad s.t.\ \mathbf{H}^\top \mathbf{H} = \mathbf{I}_k, \tag{5}$$

where $\mathbf{I}_k$ represents the identity matrix with the size of $k \times k$. Additionally, for ease of expression, $\mathbf{H}$ is referred to as the partition matrix in our paper.

Considering that the data distribution can not be sufficiently depicted in the original feature space for some complex datasets, a feature mapping $\varphi(\cdot)$ is introduced to map samples into a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$, i.e., $\phi_i = \varphi(\mathbf{x}_i)$. Since the mapping $\varphi(\cdot)$ is implicitly defined in most cases, we often construct the kernel matrix as $\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) = \phi(x_i)^\top \phi(x_j)$, and the corresponding objective function of k-means is formulated as [41], [42]:

$$\min_{\mathbf{H}} \mathrm{Tr}(\mathbf{K}(\mathbf{I}_n - \mathbf{H}\mathbf{H}^\top)) \quad s.t.\ \mathbf{H}^\top \mathbf{H} = \mathbf{I}_k. \tag{6}$$

Although Eq. (6) has demonstrated satisfactory performance in data clustering, the quality of the selected kernels greatly affects the final clustering results, and determining the optimal kernel in realistic scenarios is challenging. To address this issue, several kernel k-means based multi-view clustering algorithms have been proposed, based on the assumption that the optimal kernel is represented by a combination of pre-defined base kernels. This topic will be further discussed in the next section.

2.2 Multiple Kernel K-means Clustering

Given an optimal kernel $\mathbf{K}_\gamma = \sum_{i=1}^{v} \gamma_i^2 \mathbf{K}^{(i)}$, the multiple kernel version of objective function in Eq. (6) can be rewritten

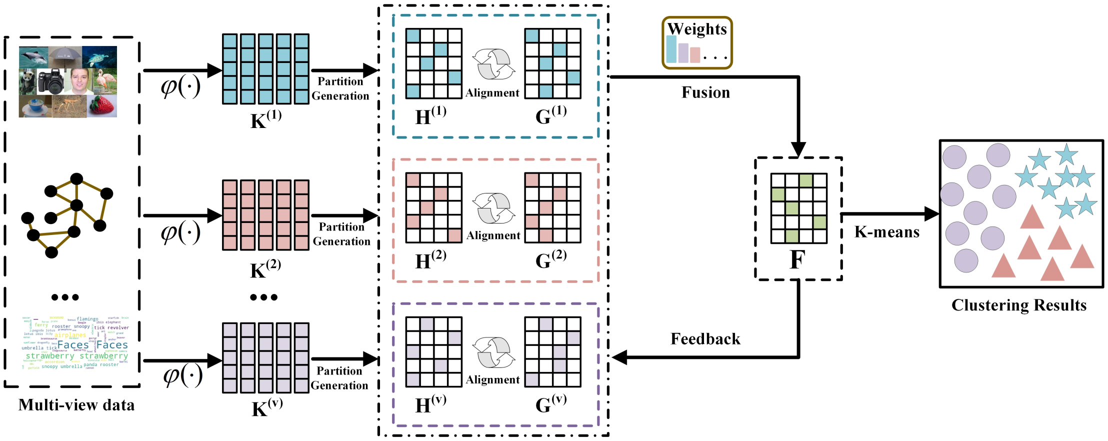


Fig. 1. Framework of FAMKKM. Firstly, according to the feature mapping function $\varphi(\cdot)$, the kernel matrices $\{\mathbf{K}^{(i)}\}_{i=1}^{v}$ are constructed from the multi-view data. Secondly, approximated partition matrices $\{\mathbf{H}^{(i)}\}_{i=1}^{v}$ and $\{\mathbf{G}^{(i)}\}_{i=1}^{v}$ are simultaneously generated from each kernel and then the proposed method is adopted to obtain the consensus partition matrix $\mathbf{F}$ with fusing these approximated partition matrices. Finally, the k-means clustering algorithm is applied on $\mathbf{F}$ to produce the final clustering results.

as:

$$\begin{aligned}&\min_{\mathbf{H},\gamma} \mathrm{Tr}(\mathbf{K}_{\gamma}(\mathbf{I}_n - \mathbf{H}\mathbf{H}^{\top}))\\ &s.t.\ \mathbf{H}^{\top}\mathbf{H} = \mathbf{I}_k, \gamma^{\top}\mathbf{1}_v = 1, \gamma_v \geq 0, \forall v.\end{aligned} \tag{7}$$

For Eq. (7), a two-step alternate optimization algorithm is usually adopted to obtain the unified kernel matrix $\mathbf{H}$, on which the standard k-means is performed to generate the final clustering results [43].

Based on the objective function outlined above, several variants have been developed to enhance clustering performance, including those presented in [26], [44], [45]. Although these variants demonstrate promising results, we have observed that many suffer from cubic complexity, primarily due to eigen-decomposition on the unified kernel matrix. As a result, the following section will propose a straightforward yet effective method for reducing the computational complexity associated with eigen-decomposition.

3 Proposed Method

3.1 Notations

In this section, we briefly introduce the notations adopted in this paper. To differentiate between variables, matrices, and vectors are represented by bold uppercase and lowercase letters, respectively, while scalars are denoted in non-bold italic font. Let $\mathbf{M} \in \mathbb{R}^{n \times d}$ be a matrix with n samples and d features, where $\mathrm{Tr}(\mathbf{M})$ and $\mathbf{M}^{\top}$ denote its trace and transpose, respectively, and M_{ij} represents its (i, j)-th element. The identity matrix of size $n \times n$ is represented by $\mathbf{I}_n$, and m_i denotes the i-th element of vector $\mathbf{m}$.

3.2 Overview

The proposed FAMKKM follows the flowchart shown in Fig. 1. Firstly, the original multi-view data matrices undergo a transformation into kernel matrices, denoted as $\{\mathbf{K}^{(i)}\}_{i=1}^{v}$, facilitated by the kernel space mapping $\varphi(\cdot)$, commonly referred to as kernel generation. Secondly, we concurrently generate approximated partition matrices $\{\mathbf{H}^{(i)}\}_{i=1}^{v}$ and $\{\mathbf{G}^{(i)}\}_{i=1}^{v}$ for each kernel, derived directly from their corresponding kernel matrices. Subsequently, the proposed fusion method is employed to consolidate these approximated partition matrices, ultimately yielding the unified partition matrix $\mathbf{F}$. Finally, we employ the standard k-means clustering algorithm on the consensus partition matrix $\mathbf{F}$ to derive the final clustering results.

3.3 Approximated Multiple Kernel K-means Clustering

As discussed above, the computational complexity of multiple kernel clustering methods primarily lies in conducting eigen-decomposition on the original kernel matrices, incurring a computational complexity of $\mathcal{O}(n^3)$. To this end, we develop a novel strategy to alleviate this computational overhead on the basis of Theorem 1. Specifically, we introduce two approximated partition matrices, denoted as $\mathbf{H}$ and $\mathbf{G}$, to address the computational intricacies associated with extant multiple kernel clustering techniques. Instead of relying solely on the original single partition matrix, our proposed method incorporates these two approximated partition matrices throughout the entire optimization process. This strategic shift serves to mitigate the complexity of partition matrix optimization. Notably, this reduction is particularly pronounced when the parameter k, representing the number of clusters, is considerably smaller than n. In this context, the computational burden is notably lightened, as only an $n \times k$ matrix size is involved in the eigen-decomposition, as opposed to the original $n \times n$ kernel matrix.

Theorem 1. *Minimizing Eq. (6) is conceptually equivalent to maximize* $\mathrm{Tr}(\mathbf{H}^{\top}\mathbf{K}\mathbf{G})$ *with constraints* $\mathbf{H}^{\top}\mathbf{H} = \mathbf{G}^{\top}\mathbf{G} = \mathbf{I}_k$.

Proof. Suppose the semi-positive definite kernel matrix $\mathbf{K}$ can be represented by $\mathbf{K} = \mathbf{U}\sum\mathbf{U}^{\top}$, according to singular value decomposition (SVD), the following inequality can be obtained:

$$\begin{aligned}&\mathrm{Tr}(\mathbf{H}^{\top}\mathbf{K}\mathbf{G})\\ &= \mathrm{Tr}(\mathbf{H}^{\top}\mathbf{U}\sum\mathbf{U}^{\top}\mathbf{G})\\ &\leq \tfrac{1}{2}(||\mathbf{H}^{\top}\mathbf{U}\sum^{\frac{1}{2}}||_F^2 + ||\mathbf{G}^{\top}\mathbf{U}\sum^{\frac{1}{2}}||_F^2)\\ &= \tfrac{1}{2}(\mathrm{Tr}(\mathbf{H}^{\top}\mathbf{K}\mathbf{H}) + \mathrm{Tr}(\mathbf{G}^{\top}\mathbf{K}\mathbf{G}))\\ &\leq \tfrac{1}{2}(\mathrm{Tr}(\mathbf{H}^{\top}\mathbf{K}\mathbf{H}) + \sum_{i=1}^{k}\sigma_i)\end{aligned} \tag{8}$$

where $\{\sigma_i\}_{i=1}^k$ are the top k largest eigenvalues of kernel matrix $\mathbf{K}$. When $\mathbf{H} = \mathbf{G}$, the inequality holds. Noticed that if we consider the comprehensive proof of Theorem 1 purely from a mathematical standpoint, it is essential to acknowledge that the minimization of Eq. (6) may not be entirely conceptually equivalent to the maximization of $\mathrm{Tr}(\mathbf{H}^\top\mathbf{K}\mathbf{G})$ under the constraints $\mathbf{H}^\top\mathbf{H} = \mathbf{G}^\top\mathbf{G} = \mathbf{I}_k$. However, within the context of our proposed method, where partition matrices $\mathbf{H}$ and $\mathbf{G}$ are both introduced to represent the same original partition, it becomes imperative that they maintain conceptual equivalence. This represents a foundational constraint that must be rigorously satisfied in our proposed method. To ensure this strict obeys the conceptual equivalence, our proposed method exclusively considers scenarios where the equality condition in Eq. (8) holds. In this light, Theorem 1 stands as a reasonable and accurate representation of our proposed approach, i.e., maximizing $\mathrm{Tr}(\mathbf{H}^\top\mathbf{K}\mathbf{G})$ is conceptually equivalent to the minimize of Eq. (6). □

As stated in Theorem 1, we establish a more efficient approach for generating partitions from individual kernel matrices when compared to prior methods. Leveraging this advancement, we proceed to formulate the initial objective function as follows:

$$\begin{aligned}&\max_{\mathbf{H},\mathbf{G}} \sum_{i=1}^{v} \mathrm{Tr}(\mathbf{H}^{(i)\top}\mathbf{K}^{(i)}\mathbf{G}^{(i)}) + \lambda_1 \mathrm{Tr}(\mathbf{H}^{(i)\top}\mathbf{G}^{(i)})\\ &s.t.\ \mathbf{H}^{(i)\top}\mathbf{H}^{(i)} = \mathbf{I}_k, \mathbf{G}^{(i)\top}\mathbf{G}^{(i)} = \mathbf{I}_k.\end{aligned} \tag{9}$$

where λ_1 is a trade-off parameter introduced into the formulation. Since partition matrices $\mathbf{H}$ and $\mathbf{G}$ are both employed to represent the same original partition, it is imperative that they remain as similar as possible. To achieve this goal, we impose a maximize alignment constraint on them, as denoted by the second term in Eq. (9). Noticed that either of these matrices can be effectively employed to generate the final clustering results, as they assume equivalent roles as the original single partition matrix.

Once the base partitions $\mathbf{H}$ and $\mathbf{G}$ have been derived for all views, the question naturally arises: how can we generate an improved consensus partition for subsequent clustering tasks? The intuitive approach might involve averaging all base partitions. However, this method fails to account for the distinctions among these base partitions, particularly the significance of each kernel partition. Inspired by the previous work in [43], our proposed method adopts a straightforward yet highly effective multiple partitions fusion strategy to integrate the information of all partition matrices into a consensus one, and the specific formulation is outlined as follows:

$$\begin{aligned}&\max_{\mathbf{F},\mathbf{R},\mathbf{W},\boldsymbol{\gamma}} \mathrm{Tr}((\mathbf{F}^\top \sum_{i=1}^{v} \gamma_i \mathbf{H}^{(i)}\mathbf{R}^{(i)}) + \mathrm{Tr}(\mathbf{F}^\top \sum_{i=1}^{v} \gamma_i \mathbf{G}^{(i)}\mathbf{W}^{(i)}))\\ &s.t.\ \mathbf{R}^{(i)\top}\mathbf{R}^{(i)} = \mathbf{I}_k, \mathbf{W}^{(i)\top}\mathbf{W}^{(i)} = \mathbf{I}_k,\\ &\mathbf{F}^\top\mathbf{F} = \mathbf{I}_k, \sum_{i=1}^{v} {\gamma_i}^2 = 1, \gamma_i \geq 0.\end{aligned} \tag{10}$$

For Eq. (10), the critical problem is how to efficiently measure the difference between these partition matrices and the consensus one, such that the complementary information of each partition matrix can be fully captured and preserved. Accordingly, two partition matrices $\mathbf{R}$ and $\mathbf{W}$ are employed to ensure that each partition matrix is closer to the consensus one by encoding the difference between them. Furthermore, considering that each partition contributes to the consensus one differently, weight coefficient γ is introduced to store the prior information in the proposed method.

It is well-recognized that partition matrices are derived from the original kernel matrices, a process that inherently entails the loss of some information. In acknowledgment of this inherent limitation, we undertake the endeavor of integrating partition information and kernel information within a unified framework, facilitating mutual information enhancement. Consequently, the final objective function is constructed through the joint optimization of Eq. (9) and Eq. (10), as formulated below:

$$\begin{aligned}&\max_{\substack{\mathbf{H},\mathbf{G},\mathbf{F},\\ \mathbf{R},\mathbf{W},\boldsymbol{\gamma}}} \sum_{i=1}^{v} \mathrm{Tr}(\mathbf{H}^{(i)\top}\mathbf{K}^{(i)}\mathbf{G}^{(i)}) + \lambda_1 \sum_{i=1}^{v} \mathrm{Tr}(\mathbf{H}^{(i)\top}\mathbf{G}^{(i)})\\ &+\lambda_2 \mathrm{Tr}((\mathbf{F}^\top \sum_{i=1}^{v} \gamma_i \mathbf{H}^{(i)}\mathbf{R}^{(i)}) + \mathrm{Tr}(\mathbf{F}^\top \sum_{i=1}^{v} \gamma_i \mathbf{G}^{(i)}\mathbf{W}^{(i)}))\\ &s.t.\ \mathbf{H}^{(i)\top}\mathbf{H}^{(i)} = \mathbf{I}_k, \mathbf{G}^{(i)\top}\mathbf{G}^{(i)} = \mathbf{I}_k, \mathbf{R}^{(i)\top}\mathbf{R}^{(i)} = \mathbf{I}_k,\\ &\mathbf{W}^{(i)\top}\mathbf{W}^{(i)} = \mathbf{I}_k, \mathbf{F}^\top\mathbf{F} = \mathbf{I}_k, \sum_{i=1}^{v} {\gamma_i}^2 = 1, \gamma_i \geq 0.\end{aligned} \tag{11}$$

where λ_2 is also a trade-off parameter.

Similar to the conventional multiple kernel k-means clustering method, the first term within Eq. (11) characterizes the process of partition generation for each kernel, with the primary divergence is that two approximated partition matrices are introduced to significantly reduce computational costs by virtue of Theorem 1. As mentioned above, two approximated partition matrices are employed to represent the same original one, which prompts the introduction of the second term to keep the consistency between them. Following the construction of partitions associated with each kernel matrix, the task of generating the optimal consensus partition is addressed through the application of an adaptive weighted partition fusion mechanism, embodied in the final two terms of Eq. (11). Consequently, our proposed method, by employing the above strategies to formulate the overall objective function, simultaneously integrates the fusion of partition information and kernel information within a unified framework. This integration is beneficial to enhance clustering performance, surpassing the outcomes attainable through partition fusion or kernel fusion alone. Furthermore, a notable advantage of Eq. (11) lies in its capacity to significantly reduce overall computational complexity, a benefit attributed to the introduction of two approximated partition matrices. A detailed analysis of this reduction in computational complexity will be elaborated in the subsequent section.

4 Optimization

As observed in Eq. (11), the proposed method contains six variables, i.e., $\mathbf{H}$, $\mathbf{G}$, $\mathbf{F}$, $\mathbf{R}$, $\mathbf{W}$, and γ. Since all variables are coupled together in Eq. (11), it is difficult to directly solve them in one step. Thus, an iterative optimization algorithm is designed to solve it in this section, and the specific process is shown as follows.

Update $\mathbf{F}$: When the variables $\mathbf{H}$, $\mathbf{G}$, $\mathbf{R}$, $\mathbf{W}$, and γ are held constant, Eq. (11) can be rendered in a simplified form:

$$\begin{aligned}&\max_{\mathbf{F}} \mathrm{Tr}\left(\mathbf{F}^\top \lambda_2 \sum_{i=1}^{v} \gamma_i (\mathbf{H}^{(i)}\mathbf{R}^{(i)} + \mathbf{G}^{(i)}\mathbf{W}^{(i)})\right)\\ &s.t.\ \mathbf{F}^\top \mathbf{F} = \mathbf{I}_k.\end{aligned} \tag{12}$$

Update $\mathbf{H}$: When considering the fixed variables $\mathbf{F}$, $\mathbf{G}$, $\mathbf{R}$, $\mathbf{W}$, and γ, we can express Eq. (11) in the following form:

$$\begin{aligned}&\max_{\mathbf{H}} \sum_{i=1}^{v} \mathrm{Tr}\left(\mathbf{H}^{(i)\top}(\mathbf{K}^{(i)}\mathbf{G}^{(i)} + \lambda_1 \mathbf{G}^{(i)} + \lambda_2 \gamma_i \mathbf{F}\mathbf{R}^{(i)\top})\right)\\ &s.t.\ \mathbf{H}^{(i)\top}\mathbf{H}^{(i)} = \mathbf{I}_k.\end{aligned} \tag{13}$$

Update $\mathbf{G}$: With variables $\mathbf{F}$, $\mathbf{H}$, $\mathbf{R}$, $\mathbf{W}$, and γ being fixed, Eq. (11) is equivalent to:

$$\begin{aligned}&\max_{\mathbf{G}} \sum_{i=1}^{v} \mathrm{Tr}\left(\mathbf{G}^{(i)\top}(\mathbf{K}^{(i)\top}\mathbf{H} + \lambda_1 \mathbf{H}^{(i)} + \lambda_2 \gamma_i \mathbf{F}\mathbf{W}^{(i)\top})\right)\\ &s.t.\ \mathbf{G}^{(i)\top}\mathbf{G}^{(i)} = \mathbf{I}_k.\end{aligned} \tag{14}$$

Update $\mathbf{R}$: With variables $\mathbf{F}$, $\mathbf{H}$, $\mathbf{G}$, $\mathbf{W}$, and γ being fixed, Eq. (11) can be denoted as:

$$\max_{\mathbf{R}} \sum_{i=1}^{v} \mathrm{Tr}\left(\mathbf{R}^{(i)\top}(\lambda_2 \gamma_i \mathbf{H}^{(i)\top}\mathbf{F})\right)\ s.t.\ \mathbf{R}^{(i)\top}\mathbf{R}^{(i)} = \mathbf{I}_k. \tag{15}$$

Update $\mathbf{W}$: With variables $\mathbf{F}$, $\mathbf{H}$, $\mathbf{G}$, $\mathbf{R}$, and γ being fixed, Eq. (11) is reformulated as:

$$\max_{\mathbf{W}} \sum_{i=1}^{v} \mathrm{Tr}\left(\mathbf{W}^{(i)\top}(\lambda_2 \gamma_i \mathbf{G}^{(i)\top}\mathbf{F})\right)\ s.t.\ \mathbf{W}^{(i)\top}\mathbf{W}^{(i)} = \mathbf{I}_k. \tag{16}$$

The above objective functions, from Eq. (12) to Eq. (16), are all can be generalized as:

$$\max_{\mathbf{U}} \mathrm{Tr}(\mathbf{U}^\top \mathbf{V}) \quad s.t.\ \mathbf{U}^\top \mathbf{U} = \mathbf{I}_k. \tag{17}$$

For example, if we define $\mathbf{V} = \lambda_2 \sum_{i=1}^{v} \gamma_i (\mathbf{H}^{(i)}\mathbf{R}^{(i)} + \mathbf{G}^{(i)}\mathbf{W}^{(i)})$ and $\mathbf{U} = \mathbf{F}$, the objective function of Eq. (17) is equivalent to the problem (12). When we replace $\mathbf{U}$ and $\mathbf{V}$ with the other variables, a similar optimization problem can also be obtained, thereby the details of them are not introduced here. In the context of Eq. (17), the optimal solution for $\mathbf{U}$ can be readily obtained by performing a singular value decomposition (SVD) on the matrix $\mathbf{V}$. To be specific, assuming that the matrix $\mathbf{V}$ takes the form $\mathbf{V} = \mathbf{M}\sum\mathbf{N}^\top$, where $\mathbf{M} \in \mathbb{R}^{n\times k}$, $\sum \in \mathbf{R}^{k\times k}$, $\mathbf{N} \in \mathbb{R}^{k\times k}$, then the closed-form solution for $\mathbf{U}$ can be expressed as $\mathbf{U} = \mathbf{M}\mathbf{N}^\top$. The specific proof process can refer to [46].

Update $\boldsymbol{\gamma}$: With variables $\mathbf{F}$, $\mathbf{H}$, $\mathbf{G}$, $\mathbf{R}$, and $\mathbf{W}$ being fixed, Eq. (11) can be expressed as:

$$\max_{\boldsymbol{\gamma}} \sum_{i=1}^{v} \gamma_i \beta_i \quad s.t.\ \sum_{i=1}^{v} {\gamma_i}^2 = 1, \gamma_i \geq 0. \tag{18}$$

where $\beta_i = \mathrm{Tr}(\mathbf{F}^\top(\mathbf{H}^{(i)}\mathbf{R}^{(i)} + \mathbf{G}^{(i)}\mathbf{W}^{(i)}))$. According to [43], the optimal solution of Eq. (18) is $\gamma_i = \frac{\beta_i}{\sqrt{\sum_{i=1}^{v} {\beta_i}^2}}$.

In a nutshell, the details for solving FAMKKM are outlined in Algorithm 1.

Algorithm 1 Fast Approximated Multiple Kernel K-means

```
Input: base kernel matrices $\{\mathbf{K}^{(i)}\}_{i=1}^{v}$, parameter $\lambda_1$ and $\lambda_2$.
Initialize $\{\mathbf{H}^{(i)}\}_{i=1}^{v}$, $\{\mathbf{G}^{(i)}\}_{i=1}^{v}$, $\{\mathbf{R}^{(i)}\}_{i=1}^{v}$, $\{\mathbf{W}^{(i)}\}_{i=1}^{v}$, $t = 1$.
while not converged do
    Update $\mathbf{F}$ via solving Eq. (12).
    Update $\{\mathbf{H}^{(i)}\}_{i=1}^{v}$ via solving Eq. (13).
    Update $\{\mathbf{G}^{(i)}\}_{i=1}^{v}$ via solving Eq. (14).
    Update $\{\mathbf{R}^{(i)}\}_{i=1}^{v}$ via solving Eq. (15).
    Update $\{\mathbf{W}^{(i)}\}_{i=1}^{v}$ via solving Eq. (16).
    Update $\{\boldsymbol{\gamma}\}_{i=1}^{v}$ via solving Eq. (18).
    $t = t + 1$.
end while
Output: Optimal partition $\mathbf{F}$.
```

4.1 Convergence Analysis

As aforementioned, the proposed method is not a joint convex problem and we adopt an alternate optimization algorithm to solve it. Based on the above optimization process, the optimal solution of each sub-problem can be easily obtained, thereby the whole algorithm converges obviously. The detailed convergence analysis of each sub-problem is shown as follows:

Updating $\mathbf{F}$. Suppose $\mathbf{Y} = \sum_{i=1}^{v} \lambda_2 \gamma_i \mathbf{H}^{(i)}\mathbf{R}^{(i)}$, according to section 4.4 in [47], we can obtain the inequality as $\mathrm{Tr}(\mathbf{F}^\top\mathbf{Y}) \leq \frac{1}{2}(\mathrm{Tr}(\mathbf{F}^\top\mathbf{F}) + \mathrm{Tr}(\mathbf{Y}^\top\mathbf{Y})) \leq \frac{\lambda_2}{2}(k + kv^2)$, and the inequality also holds when $\mathbf{Y} = \sum_{i=1}^{v} \lambda_2 \gamma_i \mathbf{G}^{(i)}\mathbf{W}^{(i)}$. Thus, the sub-problem in Eq. (12) is upper-bounded. Additionally, since the first-order derivation of Eq. (12) with respect to $\mathbf{F}$ is fixed, the whole objective function is monotonic. Accordingly, this sup-problem can reach convergence.

Updating $\mathbf{H}$, $\mathbf{G}$, $\mathbf{R}$, $\mathbf{W}$, $\boldsymbol{\gamma}$. These sup-problems are similar to the problem in Eq. (12), and we do not introduce them in detail here.

In summary, when the expression of Eq. (11) is simplified as:

$$\max_{\substack{\mathbf{H},\mathbf{G},\mathbf{F},\\ \mathbf{R},\mathbf{W},\boldsymbol{\gamma}}} \Theta(\mathbf{H}, \mathbf{G}, \mathbf{R}, \mathbf{W}, \boldsymbol{\gamma}, \mathbf{F}), \tag{19}$$

the following inequality holds:

$$\begin{aligned}&\Theta(\mathbf{H}_t, \mathbf{G}_t, \mathbf{R}_t, \mathbf{W}_t, \boldsymbol{\gamma}_t, \mathbf{F}_t)\\ &\leq \Theta(\mathbf{H}_{t+1}, \mathbf{G}_{t+1}, \mathbf{R}_{t+1}, \mathbf{W}_{t+1}, \boldsymbol{\gamma}_{t+1}, \mathbf{F}_{t+1})\end{aligned} \tag{20}$$

where superscript t represents the t-th iteration in the whole optimization. Therefore, Eq. (11) monotonically increases at each iteration. Additionally, since each independent sub-problem is upper-bounded, the whole objective function is upper-bounded. Accordingly, the proposed method can reach convergence.

4.2 Time Complexity Analysis

The proposed method contains six variables, i.e., $\mathbf{F}$, $\mathbf{H}$, $\mathbf{G}$, $\mathbf{R}$, $\mathbf{W}$, and $\boldsymbol{\gamma}$. Thus, the complexity of FAMKKM mainly lies in solving them. According to the above optimization processes, we can find that the complexity of updating $\mathbf{F}$, $\mathbf{H}$, $\mathbf{G}$, $\mathbf{R}$ and $\mathbf{W}$ are all $\mathcal{O}(nk^2)$ since the optimal solution of each variable is generated by adopting eigen-decomposition on the matrix with the size of $n \times k$. Additionally, for updating $\boldsymbol{\gamma}$, the time complexity is $\mathcal{O}(v)$. Therefore, the total computational complexity of the proposed method for each iteration is $\mathcal{O}(6nk^2 + v)$, which is linear with the number of samples n.

TABLE 1
Summary of six benchmark multiple kernel datasets.

Dataset	#Samples	#Views	#Classes
Pollen	249	12	11
Caltech102-10	1020	48	102
Caltech102-20	2040	48	102
Caltech102-30	3060	48	102
flower102	8189	4	102
ALOI-100	10800	4	100

5 Experiments

In this section, extensive experiments are conducted to verify the effectiveness of the proposed method.

5.1 Datasets

Similar to [39], we evaluate the proposed method FAMKKM on six widely used multiple kernel datasets, including Pollen [48], Caltech102-10[1], Caltech102-20[1], Caltech102-30[1], flower102[2], and ALOI-100[3], and the detailed information about these datasets are shown in Table 1.

5.2 Compared Methods

In the experiments, we compare the proposed method with the following multiple kernel clustering methods to verify its effectiveness and efficiency.

AMKKM: Average Kernel k-means.

MKKM [49]: Multiple Kernel Fuzzy Clustering.

ONKC [27]: Optimal Neighborhood Kernel Clustering with Multiple Kernels.

MKMR [24]: Multiple Kernel k-means Clustering with Matrix-induced Regularization.

MKC_NKSS [50]: Multiple Kernel Clustering with Neighbor-kernel Subspace Segmentation.

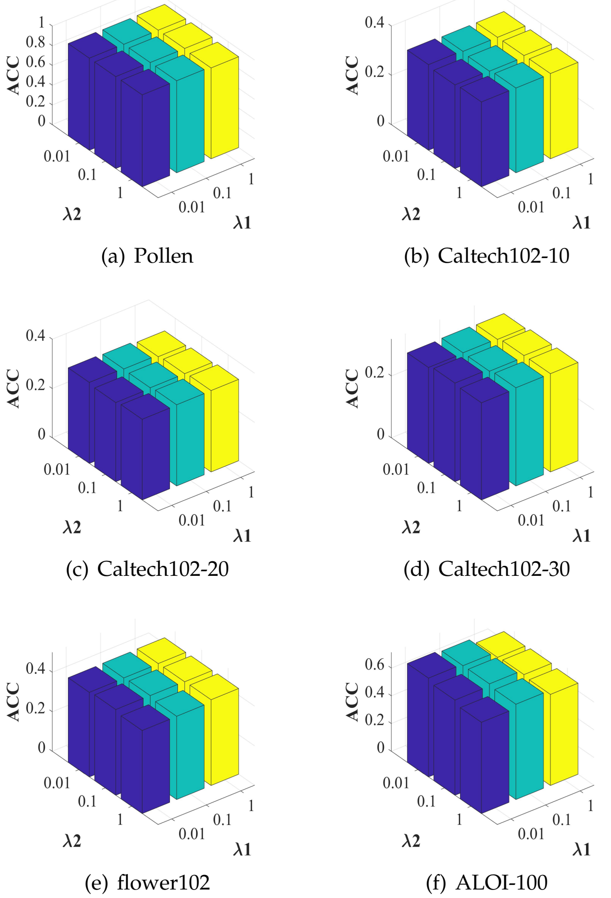
SMSC [51]: A Spectral Clustering with Self-weighted Multiple Kernel Learning Method for Single-cell RNA-seq Data.

SimpleMKKM [52]: SimpleMKKM: Simple Multiple K-ernel k-means.

MKKM-SR [53]: Multiple Kernel K-Means Clustering with Simultaneous Spectral Rotation.

5.3 Experimental Settings

For all compared methods, we have downloaded their public source codes from the corresponding websites, and the hyper-parameters involved in their methods are tuned to make them achieve optimal clustering results in the experiments. Additionally, for the proposed method FAMKKM, it contains two parameters, i.e., λ_1 and λ_2. Since the optimal parameter values are difficult to determine for each dataset, a grid-searching strategy is adopted in the experiments. Specifically, we tune λ_1 and λ_2 in a range of $[0.01, 0.1, 1]$, respectively. Furthermore, to effectively evaluate the performance of all compared methods on the above-mentioned multiple kernel datasets, seven clustering metrics are selected in the experiments, including F1 measure (Fscore), Precision, Recall, Normalized Mutual Information (NMI), Adjusted Rand Index (ARI), Accuracy (ACC), and Purity. For all competitors, we repeat them 50 times with random initialization to mitigate the impact of k-means on the final clustering results, and the corresponding averages and standard deviation of the above metrics are reported in Table 2. Since all methods use Matlab in their experiments, our experiments are also conducted on a PC with an Intel Core-i7-7700 CPU and 24GB RAM, Matlab R2020a.



(a) Pollen (b) Caltech102-10

(c) Caltech102-20 (d) Caltech102-30

(e) flower102 (f) ALOI-100

Fig. 3. Parameter sensitivity of the proposed method in terms of ACC.

5.4 Results and Analysis

In Table 2, the best clustering results on each dataset are bolded, and N/A denotes out-of-memory failure. According to the results, we can obtain the following observations.

- Across all evaluation metrics, FAMKKM consistently demonstrates superior performance when compared to alternative methods, particularly outperforming existing MKKM algorithms. For instance, on the ALOI-100 dataset, FAMKKM exhibits notable improvements, surpassing the second-best method (MKMR) by 4.87% and 3.61% in terms of F-score and ACC, respectively. These performance differentials are consistently observed across the various datasets listed in Table 2, presenting the efficacy of the

[1] http://www.vision.caltech.edu/archive.html
[2] http://www.robots.ox.ac.uk/vgg/data/flowers/
[3] http://elki.dbs.ifi.lmu.de/wiki/DataSets/MultiView

TABLE 2
Clustering results on six benchmark multiple kernel datasets in terms of seven metrics. N/A denotes out-of-memory failure. The best result in terms of each metric is bolded.

Datasets	Methods	Fscore	Precision	Recall	NMI	ARI	ACC	Purity
Pollen	AMKKM	93.25±0.39	92.57±0.16	93.93±0.63	94.47±0.33	92.46±0.43	92.05±0.32	95.30±0.19
	MKKM	93.77±0.88	91.89±1.34	95.73±0.38	95.40±0.44	93.03±0.99	92.53±0.85	95.74±0.63
	ONKC	**94.85±0.00**	93.73±0.15	96.00±0.00	95.40±0.21	**94.25±0.00**	93.21±0.13	96.39±0.00
	MKMR	93.78±0.75	**94.62±1.08**	94.45±0.63	94.74±0.33	93.06±0.84	92.41±0.64	95.50±0.17
	MKC_NKSS	81.27±1.57	72.10±2.91	93.23±1.13	89.67±0.37	78.78±1.82	83.29±1.65	84.42±1.27
	SMSC	91.54±2.16	87.67±4.25	97.88±0.13	95.08±0.79	90.48±2.44	91.97±1.69	93.33±2.30
	SimpleMKKM	83.44±1.26	79.56±2.28	87.77±1.62	89.81±0.63	81.41±1.43	84.62±2.09	86.91±1.19
	MKKM-SR	92.57±0.00	91.12±0.00	94.73±0.00	94.11±0.00	91.70±0.00	91.57±0.00	94.78±0.00
	Ours	94.84±0.00	93.66±0.21	**96.13±0.11**	**95.62±0.00**	94.24±0.00	**93.41±0.51**	**96.39±0.00**
Caltech102-10	AMKKM	15.81±0.53	14.88±0.51	16.87±0.60	62.35±0.39	15.02±0.54	31.91±0.55	33.88±0.55
	MKKM	6.05±0.46	4.85±0.45	8.03±0.49	53.45±0.63	5.00±0.48	21.05±0.87	22.59±0.89
	ONKC	16.20±0.84	15.17±0.75	17.40±0.98	62.54±0.42	15.41±0.84	32.51±1.09	34.47±0.89
	MKMR	17.02±0.69	15.96±0.65	18.23±0.75	63.01±0.72	16.23±0.70	33.26±0.83	35.32±1.42
	MKC_NKSS	12.96±0.55	10.33±0.71	18.49±0.89	59.64±0.40	11.98±0.58	27.84±0.54	29.34±0.45
	SMSC	7.21±0.49	4.49±0.37	**20.85±1.20**	51.55±0.78	5.87±0.52	24.52±0.45	28.11±0.48
	SimpleMKKM	16.93±1.31	15.86±1.34	18.15±1.28	63.05±0.87	16.13±1.33	33.28±1.43	35.25±1.26
	MKKM-SR	4.45±0.00	13.47±0.00	15.58±0.00	61.67±0.00	13.63±0.00	32.25±0.00	34.51±0.00
	Ours	**18.13±0.69**	**16.94±0.77**	19.51±0.63	**63.89±0.47**	**17.35±0.70**	**34.87±0.94**	**37.16±0.91**
Caltech102-20	AMKKM	16.70±0.68	15.80±0.67	17.70±0.72	54.69±0.48	15.87±0.69	29.84±0.89	31.70±0.89
	MKKM	6.57±0.33	5.60±0.37	7.95±0.33	44.22±0.42	5.53±0.35	17.88±0.48	19.51±0.38
	ONKC	16.84±0.77	15.94±0.75	17.85±0.82	54.76±0.53	16.01±0.78	29.96±0.89	31.75±0.66
	MKMR	18.25±0.62	17.24±0.56	19.40±0.71	56.23±0.34	17.44±0.63	31.85±0.68	33.86±0.60
	MKC_NKSS	12.88±0.50	10.42±0.33	17.05±0.75	50.91±0.38	11.86±0.52	24.13±0.67	25.66±0.66
	SMSC	4.64±0.28	2.62±0.18	**21.94±1.61**	37.68±0.82	3.04±0.30	15.88±0.59	18.92±0.75
	SimpleMKKM	18.06±0.63	17.07±0.55	19.19±0.76	56.07±0.57	17.25±0.63	31.60±0.77	33.53±0.79
	MKKM-SR	14.94±0.00	14.28±0.00	15.66±0.00	53.49±0.00	14.10±0.00	29.22±0.00	31.08±0.00
	Ours	**19.06±0.62**	**18.10±0.59**	20.13±0.68	**56.74±0.54**	**18.26±0.63**	**32.77±0.89**	**34.90±0.99**
Caltech102-30	AMKKM	16.62±0.57	15.73±0.54	17.62±0.63	50.61±0.61	15.78±0.58	28.90±0.75	30.72±0.88
	MKKM	6.87±0.21	6.01±0.24	8.03±0.29	39.41±0.36	5.85±0.22	16.28±0.48	17.68±0.33
	ONKC	16.59±0.59	15.79±0.58	17.58±0.67	50.59±0.50	15.75±0.60	28.85±0.90	30.84±0.51
	MKMR	17.83±0.45	16.87±0.44	18.91±0.49	52.00±0.36	17.00±0.45	31.31±0.65	32.84±0.55
	MKC_NKSS	12.91±0.41	11.09±0.33	15.73±0.19	47.14±0.37	11.93±0.43	23.37±0.33	24.97±0.39
	SMSC	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	SimpleMKKM	17.87±0.75	16.93±0.77	18.93±0.74	51.98±0.60	17.04±0.76	30.54±0.76	32.59±0.67
	MKKM-SR	14.35±0.00	13.76±0.00	15.01±0.00	48.60±0.00	13.50±0.00	27.45±0.00	28.99±0.00
	Ours	**18.95±0.52**	**17.97±0.54**	**20.05±0.53**	**53.17±0.29**	**18.13±0.52**	**31.87±0.90**	**34.08±0.87**
flower102	AMKKM	16.59±0.61	18.36±0.73	15.13±0.52	46.55±0.37	15.62±0.62	27.23±0.77	32.37±0.67
	MKKM	12.98±0.32	14.46±0.39	11.78±0.27	43.12±0.16	11.98±0.32	22.43±0.40	27.87±0.49
	ONKC	26.55±0.34	28.64±0.46	24.74±0.31	57.14±0.44	25.68±0.34	40.17±0.86	46.34±0.87
	MKMR	26.59±0.53	28.59±0.61	24.85±0.49	57.20±0.42	25.72±0.53	40.27±0.78	46.52±0.85
	MKC_NKSS	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	SMSC	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	SimpleMKKM	28.92±0.81	31.25±1.07	26.92±0.63	**58.82±0.39**	28.08±0.82	42.29±1.03	48.41±0.45
	MKKM-SR	17.08±0.00	18.99±0.00	15.52±0.00	47.07±0.00	16.13±0.00	29.03±0.00	34.46±0.00
	Ours	**30.37±0.66**	**33.57±0.87**	**27.73±0.57**	57.67±0.46	**29.56±0.67**	**42.84±0.98**	**49.21±0.84**
ALOI-100	AMKKM	51.93±1.60	47.84±2.14	56.82±0.97	78.42±0.51	51.40±1.63	65.22±1.67	66.99±1.43
	MKKM	3.41±0.00	2.98±0.00	3.97±0.00	25.69±0.18	2.30±0.00	6.70±0.11	7.56±0.00
	ONKC	54.42±1.22	50.83±1.66	58.59±1.06	79.15±0.57	53.93±1.24	66.79±1.29	68.55±1.65
	MKMR	55.10±1.20	51.80±1.47	58.87±1.25	79.15±0.67	54.62±1.22	67.63±1.70	69.13±1.54
	MKC_NKSS	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	SMSC	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	SimpleMKKM	52.24±0.89	48.19±1.25	56.91±0.86	78.48±0.42	51.69±0.90	65.69±1.33	67.50±1.09
	MKKM-SR	32.35±0.00	31.91±0.00	32.80±0.00	62.86±0.00	31.66±0.00	46.14±0.00	48.66±0.00
	Ours	**59.97±0.84**	**56.42±1.27**	**64.63±0.63**	**81.60±0.33**	**59.55±0.85**	**71.24±1.21**	**73.21±0.95**

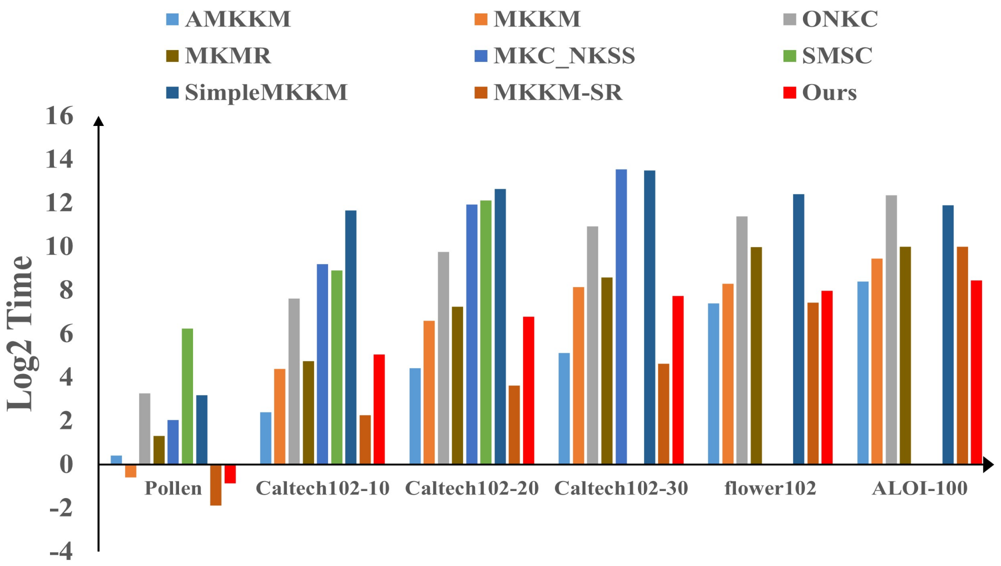


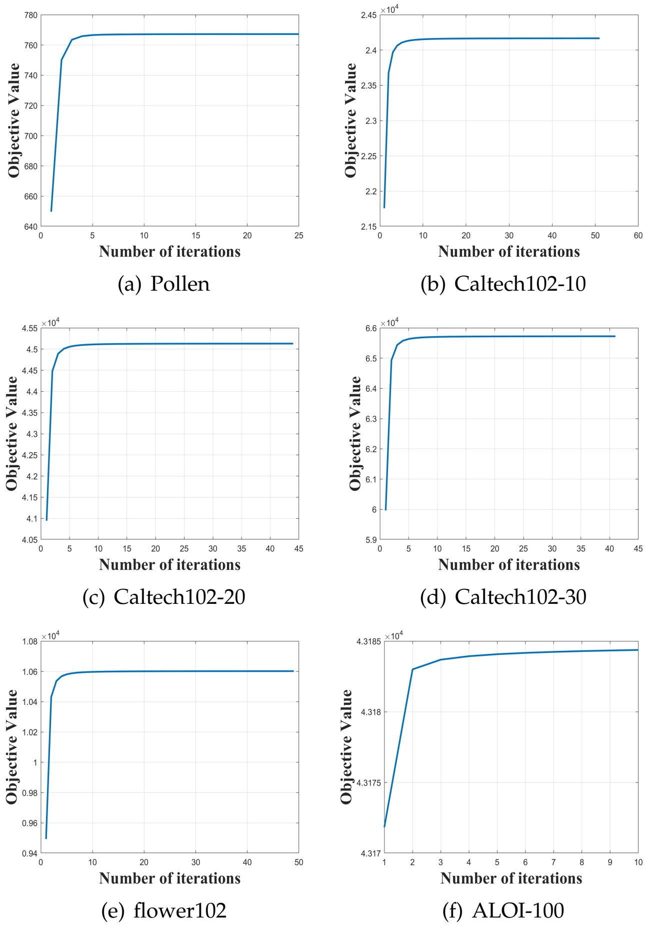
Fig. 2. Running Time comparison of different methods on all multiple kernel datasets.

TABLE 3
Clustering performance of different level information fusion on six benchmark datasets. The best result in terms of each metric is bolded.

Datasets	Methods	Fscore	Precision	Recall	NMI	ARI	ACC	Purity	Time(s)
Pollen	MKKM_PF	94.54±0.38	93.49±0.15	95.60±0.62	95.39±0.32	93.89±0.42	93.09±0.41	96.27±0.19	2.09
	MKKM_KF	93.77±0.88	91.89±1.34	95.73±0.38	95.40±0.44	93.03±0.99	92.53±0.85	95.74±0.63	1.12
	Ours	**94.84±0.00**	**93.66±0.21**	**96.13±0.11**	**95.62±0.00**	**94.24±0.00**	**93.41±0.51**	**96.39±0.00**	0.55
Caltech102-10	MKKM_PF	17.53±0.95	16.38±0.93	18.86±1.00	63.61±0.56	16.75±0.96	34.04±0.93	36.60±0.90	76.10
	MKKM_KF	6.05±0.46	4.85±0.45	8.03±0.49	53.45±0.63	5.00±0.48	21.05±0.87	22.59±0.89	69.51
	Ours	**18.13±0.69**	**16.94±0.77**	**19.51±0.63**	**63.89±0.47**	**17.35±0.70**	**34.87±0.94**	**37.16±0.91**	32.86
Caltech102-20	MKKM_PF	18.32±1.08	17.26±1.08	19.52±1.09	56.24±0.59	17.50±1.10	32.25±1.08	34.40±0.96	209.23
	MKKM_KF	6.57±0.33	5.60±0.37	7.95±0.33	44.22±0.42	5.53±0.35	17.88±0.48	19.51±0.38	288.48
	Ours	**19.06±0.62**	**18.10±0.59**	**20.13±0.68**	**56.74±0.54**	**18.26±0.63**	**32.77±0.89**	**34.90±0.99**	109.96
Caltech102-30	MKKM_PF	18.08±0.78	17.14±0.79	19.13±0.78	52.52±0.66	17.25±0.79	30.83±1.13	33.08±1.14	346.85
	MKKM_KF	6.87±0.21	6.01±0.24	8.03±0.29	39.41±0.36	5.85±0.22	16.28±0.48	17.68±0.33	744.01
	Ours	**18.95±0.52**	**17.97±0.54**	**20.05±0.53**	**53.17±0.29**	**18.13±0.52**	**31.87±0.90**	**34.08±0.87**	213.26
flower102	MKKM_PF	30.32±1.18	33.45±1.47	27.72±0.97	57.65±0.45	29.51±1.20	42.78±1.37	**49.28±0.65**	506.26
	MKKM_KF	12.98±0.32	14.46±0.39	11.78±0.27	43.12±0.16	11.98±0.32	22.43±0.40	27.87±0.49	883.92
	Ours	**30.37±0.66**	**33.57±0.87**	**27.73±0.57**	**57.67±0.46**	**29.56±0.67**	**42.84±0.98**	49.21±0.84	251.90
ALOI-100	MKKM_PF	54.69±1.90	49.41±2.49	61.27±1.09	80.42±0.57	54.18±1.93	68.81±1.37	70.83±0.97	449.82
	MKKM_KF	3.41±0.00	2.98±0.00	3.97±0.00	25.69±0.18	2.30±0.00	6.70±0.11	7.56±0.00	702.02
	Ours	**59.97±0.84**	**56.42±1.27**	**64.63±0.63**	**81.60±0.33**	**59.55±0.85**	**71.24±1.21**	**73.21±0.95**	347.53

proposed FAMKKM and its associated optimization algorithm

- An additional major advantage of our proposed method is the interactive fusion of partition information and kernel information, a feature that distinguishes it from existing multiple kernel clustering algorithms. This fusion capability constitutes a primary reason for FAMKKM's consistently superior performance across all benchmark datasets. In contrast, other multiple kernel clustering methods, such as SimpleMKKM and MKKM-SR, either focus solely on fusing latent information at the partition level or the kernel matrix level. Our experimental results affirm that our proposed approach significantly outperforms these other methods. This advantage stems from the inherent qualities of partition-level information, which tends to exhibit less noise and feature redundancy, and kernel matrix information, which is often more informative. By exploiting both forms of information, our approach efficiently generates an improved unified kernel partition, thereby enhancing the final clustering performance.



(a) Pollen

(b) Caltech102-10

(c) Caltech102-20

(d) Caltech102-30

(e) flower102

(f) ALOI-100

Fig. 4. Convergence curves of FAMKKM on all multiple kernel datasets.

- As seen in Fig. 2, our proposed method also has the advantage in terms of running time efficiency compared to other methods. This advantage can be attributed to the introduction of two approximated partition matrices. Like many other methods, our proposed method also needs to conduct SVD on the original kernel matrix to generate the clustering partition in each iteration. However, FAMKKM specifically conducts SVD on a smaller matrix with the size of $n \times k$, resulting in a substantial reduction in computational complexity, from $\mathcal{O}(n^3)$ to $\mathcal{O}(n)$. Consequently, compared with the other competitors, our proposed method is more readily applied to large-scale multiple kernel datasets in terms of clustering performance and efficiency.

5.5 Model Evaluation

This section presents ablation experiments designed to assess the impact of utilizing different levels of information on the performance of multiple kernel k-means clustering. To be specific, partition level information and the original kernel information are adopted separately to compare with the proposed method, and the corresponding clustering results on six benchmark datasets are presented in Table 3. For ease of expression, the method with only involves partition fusion is referred to as MKKM_PF, and the other one is referred to as MKKM_KF. As illustrated in Table 3, it is evident that our proposed method outperforms other techniques. This highlights the efficacy of simultaneously exploiting partition-level and kernel-level information in multiple kernel clustering. Moreover, by introducing two approximated partition matrices, our proposed method can further reduce computational complexity. Based on these clustering results, we conclude that our proposed method is a straightforward yet highly effective approach to multiple kernel clustering.

5.6 Parameter and Convergence Study

According to Eq. (11), it can be observed that the proposed method contains two parameters, i.e., λ_1 and λ_2. To further study the parameter sensitivity of FAMKKM on all multiple kernel datasets, we conduct the parameter sensitivity analysis experiments in this section, and the experimental results in terms of ACC are given in Fig. 3. As seen in the figures, we can find that the proposed method is not sensitive to two parameters on all datasets. Within the given range of parameters, FAMKKM can achieve satisfying performance.

In the aforementioned section, we have theoretically proved the convergence property of the proposed method. Now the experiments are conducted to further verify its convergence, and the corresponding results are shown in Fig. 4. As can be seen, the proposed method reaches convergence quickly, usually within several iterations, which can be attributed to the fact that we solve each sub-problem optimally.

6 Conclusion

In this paper, a simple yet effective multiple kernel k-means clustering method is developed to address the large-scale data clustering problem, namely, FAMKKM. By introducing two approximated partition matrices and coupling the partition fusion and kernel fusion into a unified framework, the computational complexity of the proposed method is significantly reduced, as well as improving the final clustering performance. Results from experiments validate the superiority of the proposed method compared with the other algorithms.

References

[1] 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.

[2] K. Liang, Y. Liu, S. Zhou, W. Tu, Y. Wen, X. Yang, X. Dong, and X. Liu, "Knowledge graph contrastive learning based on relation-symmetrical structure," *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[3] K. Liang, L. Meng, M. Liu, Y. Liu, W. Tu, S. Wang, S. Zhou, X. Liu, and F. Sun, "Reasoning over different types of knowledge graphs: Static, temporal and multi-modal," *arXiv preprint arXiv:2212.05767*, 2022.

[4] M.-S. Chen, L. Huang, C.-D. Wang, and D. Huang, "Multi-view clustering in latent embedding space," in *AAAI*, vol. 34, no. 04, 2020, pp. 3513–3520.

[5] C. Xu, Z. Guan, W. Zhao, H. Wu, Y. Niu, and B. Ling, "Adversarial incomplete multi-view clustering." in *IJCAI*, 2019, pp. 3933–3939.

[6] 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," *Briefings in Bioinformatics*, p. bbad216, 2023.

[7] J. Liu, X. Liu, Y. Yang, Q. Liao, and Y. Xia, "Contrastive multi-view kernel learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[8] J. Xu, J. Han, and F. Nie, "Discriminatively embedded k-means for multi-view clustering," in *CVPR*, 2016, pp. 5356–5364.

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

[10] 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.

[11] Z. Tao, H. Liu, S. Li, Z. Ding, and Y. Fu, "From ensemble clustering to multi-view clustering," in *IJCAI*, 2017.

[12] J. Wen, Z. Zhang, Z. Zhang, Z. Wu, L. Fei, Y. Xu, and B. Zhang, "Dimc-net: Deep incomplete multi-view clustering network," in *ACM MM*, 2020, pp. 3753–3761.

[13] Z. Li, C. Tang, X. Liu, X. Zheng, W. Zhang, and E. Zhu, "Consensus graph learning for multi-view clustering," *IEEE Transactions on Multimedia*, vol. 24, pp. 2461–2472, 2021.

[14] C. Tang, Z. Li, J. Wang, X. Liu, W. Zhang, and E. Zhu, "Unified one-step multi-view spectral clustering," *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[15] W. Liang, X. Liu, Y. Liu, C. Ma, Y. Zhao, Z. Liu, and E. Zhu, "Consistency of multiple kernel clustering," in *ICML*, 2023.

[16] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the royal statistical society. series c (applied statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[17] A. Likas, N. Vlassis, and J. J. Verbeek, "The global k-means clustering algorithm," *Pattern recognition*, vol. 36, no. 2, pp. 451–461, 2003.

[18] S. Na, L. Xumin, and G. Yong, "Research on k-means clustering algorithm: An improved k-means clustering algorithm," in *2010 Third International Symposium on intelligent information technology and security informatics*. Ieee, 2010, pp. 63–67.

[19] K. Wagstaff, C. Cardie, S. Rogers, S. Schrödl *et al.*, "Constrained k-means clustering with background knowledge," in *ICML*, vol. 1, 2001, pp. 577–584.

[20] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: Analysis and implementation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 881–892, 2002.

[21] 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*, vol. 34, no. 7, pp. 3418–3430, 2020.

[22] B. Dash, D. Mishra, A. Rath, and M. Acharya, "A hybridized k-means clustering approach for high dimensional dataset," *International Journal of Engineering, Science and Technology*, vol. 2, no. 2, pp. 59–66, 2010.

[23] M. Li, X. Liu, L. Wang, Y. Dou, J. Yin, and E. Zhu, "Multiple kernel clustering with local kernel alignment maximization," in *IJCAI*, 2016, pp. 1704–1710.

[24] X. Liu, Y. Dou, J. Yin, L. Wang, and E. Zhu, "Multiple kernel k-means clustering with matrix-induced regularization," in *AAAI*, vol. 30, no. 1, 2016.

[25] 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, 2011.

[26] Y. Lu, L. Wang, J. Lu, J. Yang, and C. Shen, "Multiple kernel clustering based on centered kernel alignment," *Pattern Recognition*, vol. 47, no. 11, pp. 3656–3664, 2014.

[27] X. Liu, S. Zhou, Y. Wang, M. Li, Y. Dou, E. Zhu, and J. Yin, "Optimal neighborhood kernel clustering with multiple kernels," in *AAAI*, vol. 31, no. 1, 2017.

[28] 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*, vol. 41, no. 10, pp. 2410–2423, 2018.

[29] X. Liu, "Incomplete multiple kernel alignment maximization for clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[30] X. Zhu, X. Liu, M. Li, E. Zhu, L. Liu, Z. Cai, J. Yin, and W. Gao, "Localized incomplete multiple kernel k-means." in *IJCAI*, 2018, pp. 3271–3277.

[31] Z. Ren and Q. Sun, "Simultaneous global and local graph structure preserving for multiple kernel clustering," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 5, pp. 1839–1851, 2020.

[32] S. Zhou, X. Liu, J. Liu, X. Guo, Y. Zhao, E. Zhu, Y. Zhai, J. Yin, and W. Gao, "Multi-view spectral clustering with optimal neighborhood laplacian matrix," in *AAAI*, vol. 34, no. 04, 2020, pp. 6965–6972.

[33] J. Wen, Z. Zhang, Z. Zhang, L. Fei, and M. Wang, "Generalized incomplete multiview clustering with flexible locality structure diffusion," *IEEE transactions on cybernetics*, vol. 51, no. 1, pp. 101–114, 2020.

[34] Z. Kang, C. Peng, Q. Cheng, and Z. Xu, "Unified spectral clustering with optimal graph," in *AAAI*, vol. 32, no. 1, 2018.

[35] X. Peng, Z. Huang, J. Lv, H. Zhu, and J. T. Zhou, "Comic: Multi-view clustering without parameter selection," in *ICML*. PMLR, 2019, pp. 5092–5101.

[36] 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 *ICML*. PMLR, 2021, pp. 6850–6859.

[37] Y. Zhang, X. Liu, S. Wang, J. Liu, S. Dai, and E. Zhu, "One-stage incomplete multi-view clustering via late fusion," in *ACM MM*, 2021, pp. 2717–2725.

[38] E. Bruno and S. Marchand-Maillet, "Multiview clustering: a late fusion approach using latent models," in *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, 2009, pp. 736–737.

[39] S. Wang, X. Liu, L. Liu, S. Zhou, and E. Zhu, "Late fusion multiple kernel clustering with proxy graph refinement," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[40] H. Zha, X. He, C. Ding, M. Gu, and H. Simon, "Spectral relaxation for k-means clustering," *NIPS*, vol. 14, 2001.

[41] R. Chitta, R. Jin, T. C. Havens, and A. K. Jain, "Approximate kernel k-means: Solution to large scale kernel clustering," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2011, pp. 895–903.

[42] 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.

[43] 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.

[44] B. Zhao, J. T. Kwok, and C. Zhang, "Multiple kernel clustering," in *ICDM*. SIAM, 2009, pp. 638–649.

[45] J. Liu, X. Liu, S. Wang, S. Zhou, and Y. Yang, "Hierarchical multiple kernel clustering," in *AAAI*, vol. 35, no. 10, 2021, pp. 8671–8679.

[46] J. Huang, F. Nie, and H. Huang, "Spectral rotation versus k-means in spectral clustering," in *AAAI*, vol. 27, no. 1, 2013, pp. 431–437.

[47] J. Wang, C. Tang, Z. Li, X. Liu, W. Zhang, E. Zhu, and L. Wang, "Hyperspectral band selection via region-aware latent features fusion based clustering," *Information Fusion*, vol. 79, pp. 162–173, 2022.

[48] A. A. Pollen, T. J. Nowakowski, J. Shuga, X. Wang, A. A. Leyrat, J. H. Lui, N. Li, L. Szpankowski, B. Fowler, P. Chen *et al.*, "Low-coverage single-cell mrna sequencing reveals cellular heterogeneity and activated signaling pathways in developing cerebral cortex," *Nature biotechnology*, vol. 32, no. 10, pp. 1053–1058, 2014.

[49] 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, 2011.

[50] S. Zhou, X. Liu, M. Li, E. Zhu, L. Liu, C. Zhang, and J. Yin, "Multiple kernel clustering with neighbor-kernel subspace segmentation," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 4, pp. 1351–1362, 2019.

[51] R. Qi, J. Wu, F. Guo, L. Xu, and Q. Zou, "A spectral clustering with self-weighted multiple kernel learning method for single-cell rna-seq data," *Briefings in Bioinformatics*, vol. 22, no. 4, p. bbaa216, 2021.

[52] X. Liu, "Simplemkkm: Simple multiple kernel k-means," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[53] J. Lu, Y. Lu, R. Wang, F. Nie, and X. Li, "Multiple kernel k-means clustering with simultaneous spectral rotation," in *ICASSP*. IEEE, 2022, pp. 4143–4147.