TFMKC: Tuning-Free Multiple Kernel Clustering Coupled With Diverse Partition Fusion

Junpu Zhang, Liang Li, Pei Zhang, Yue Liu, *Member, IEEE*, Siwei Wang, *Member, IEEE*, Changbao Zhou, Xinwang Liu, *Senior Member, IEEE*, and En Zhu

***Abstract*— Clustering is a popular research pipeline in unsupervised learning to find potential groupings. As a representative paradigm in multiple kernel clustering (MKC), late fusion-based models learn a consistent partition across multiple base kernels. Despite their promising performance, a common concern is the limited representation capacity caused by the inflexible fusion mechanism. Concretely, the representations are constrained by truncated-k Eigen-decomposition (EVD) without fully exploiting potential information. An intuitive idea to alleviate this concern is to generate a set of augmented partitions and then select the optimal partition by fine-tuning. However, this is overlimited by: 1) introducing undesired hyperparameters and dataset-related consequences; 2) neglecting rich information across diverse partitions; and 3) expensive parameter-tuning costs. To address these problems, we propose transforming the challenging problem of directly determining the optimal partition (optimal parameter) into a diverse partition fusion (parameter ensemble) problem. We design a novel flexible fusion mechanism called tuning-free multiple kernel clustering coupled with diverse partition fusion (TFMKC) by reweighting diverse partitions through optimization, achieving an optimal consensus partition by integrating diverse and complementary information rather than traditional fine-tuning, and distinguishing our work from existing methods. Extensive experiments verify that TFMKC achieves competitive effectiveness and efficiency over comparison baselines. The code can be accessed at https://github.com/ZJP/TFMKC.**

***Index Terms*— Diverse fusion, ensemble learning, late fusion kernel learning, multiview clustering.**

NOMENCLATURE

Notation	Explanation
n, k, and v	Number of clusters, samples, and views.
$\boldsymbol{\alpha} \in \mathbb{R}^{v\times 1}$	View weights.
$\boldsymbol{\beta} \in \mathbb{R}^{m\times v}$	Diverse partition weights.
$\mathbf{K}_p \in \mathbb{R}^{n\times n}$	pth base kernel.
$\mathbf{H}_p \in \mathbb{R}^{n\times k}$	pth base kernel partition.
$\mathbf{H} \in \mathbb{R}^{n\times k}$	Consensus kernel partition.
$\mathbf{V}_p \in \mathbb{R}^{k\times k}$	Permutation matrix.
$\mathbf{U}_p^{(i)} \in \mathbb{R}^{n\times d_i}$	Augmented kernel partition of $\mathbf{K}_p$.
$\mathbf{K}_p^{(i)} \in \mathbb{R}^{n\times n}$	Refined kernel from $\mathbf{U}_p^{(i)}$.

Manuscript received 6 January 2024; revised 6 June 2024; accepted 8 July 2024. This work was supported in part by the National Science Fund for Distinguished Young Scholars of China under Grant 62325604, in part by the National Key Research and Development Program of China under Grant 2021YFB0300101, in part by the National Natural Science Foundation of China under Project 62276271, and in part by China Scholarship Council under Grant 202306110001. *(Junpu Zhang and Liang Li contributed equally to this work.) (Corresponding author: Xinwang Liu.)*

Junpu Zhang, Liang Li, Pei Zhang, Yue Liu, Xinwang Liu, and En Zhu are with the School of Computer, National University of Defense Technology, Changsha 410073, China (e-mail: zhangjunpu@nudt.edu.cn; liliang1037@gmail.com; jeaninezpp@gmail.com; yueliu19990731@163.com; xinwangliu@nudt.edu.cn; enzhu@nudt.edu.cn).

Siwei Wang is with the Intelligent Game and Decision Laboratory, Beijing 100071, China (e-mail: wangsiwei13@nudt.edu.cn).

Changbao Zhou is with the College of Computer Science and Technology, Jilin University, Changchun 130012, China (e-mail: zhoucb21@mails.jlu.edu.cn).

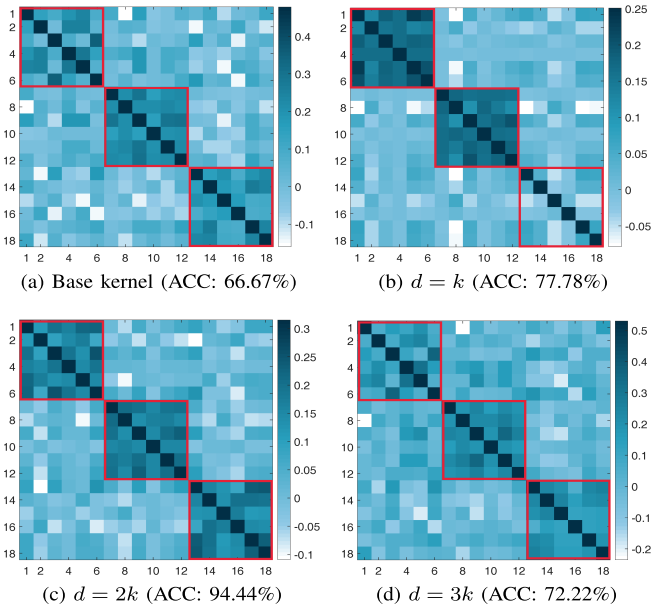
Digital Object Identifier 10.1109/TNNLS.2024.3435058

I. INTRODUCTION

CLUSTERING is a representative unsupervised learning method to explore hidden grouping structures within data [1], [2], [3], [4], [5], gained wide applications in data mining, and knowledge discovery. To tackle fast-growing multisource or multichannel data, multiple view clustering (MVC) has been an active topic in integrating diverse information [6], [7], [8], [9], [10], [11], [12]. As a representative pipeline, multiple kernel clustering (MKC) [13], [14], [15], [16] is developed to address real-world nonlinearly separable data, mapping raw features into a reproducing Kernel Hilbert Space (RKHS) [17].

Existing MKC algorithms can be classified into two branches according to the information fusion stage, i.e., kernel fusion MKC and late fusion MKC. For kernel fusion strategy, it first fuses a consistent kernel from the original multiple kernels and then computes the clustering partition [18], [19]. Multiple kernel k-means (MKKM) [20] is a representative method with extensive investigations [21], [22], [23]. Typically, SimpleMKKM [14] achieved a global optimum by formulating a min–max problem. LSWMKC [19] transferred graph construction into neighbor kernel learning to exploit local structures. In contrast, the late fusion strategy first generates base partitions from the original kernels and then fuses a consensus partition [24]. Based on this backbone, many variants are developed with various strategies [25], [26], [27]. Some recent methods explored partition approximation techniques [28] and joint feature and structural information fusion [29].

Although late fusion MKC exhibits promising performance and efficiency [24], [25], [30], many variants adopt an inflexible strategy that constrains partition representation by truncated-k Eigen-decomposition (EVD). Typically, the feature dimension of the partition is fixed at $d = k$, which means that only rank-k eigenvectors are preserved, while the rest are removed. However, a synthetic dataset shown in Fig. 1 indicates that only preserving rank-k eigenvectors is insufficient. We observe that such a way achieves



(a) Base kernel (ACC: 66.67%) (b) $d = k$ (ACC: 77.78%)

(c) $d = 2k$ (ACC: 94.44%) (d) $d = 3k$ (ACC: 72.22%)

Fig. 1. Visualizing the limited representation capacity problem in late fusion MKC, which exists in the widely used truncated-k strategy in generating base partitions. The synthetic single-view dataset $\mathbf{X}_{\text{syn}}$ consists of 18 samples generated by the normal distribution. (a) Base kernel $\mathbf{K}_{\text{syn}}$ generated by $\mathbf{X}_{\text{syn}}^{\top}\mathbf{X}_{\text{syn}}$; the metric is obtained by performing k-means on raw features $\mathbf{X}_{\text{syn}}$. (b)–(d) Refined kernel $\mathbf{K}_{\text{syn}}^{(i)} = \mathbf{U}_{\text{syn}}^{(i)}\mathbf{U}_{\text{syn}}^{(i)\top}$ computed by augmented partitions $\mathbf{U}_{\text{syn}}^{(i)}$ with truncated-d EVD, where $d \in \{k, 2k, 3k\}$; the metrics are obtained by performing k-means on $\mathbf{U}_{\text{syn}}^{(i)}$.

11.11% improvement on kernels over performing k-means on the raw features, and increasing the dimension to $2k$ further gains 16.66% improvement. However, when it increases to $3k$, the performance worsens with a negative gain of 22.22%.

The above results motivate us: *how to determine optimal partition/parameter in unsupervised late fusion MKC tasks*?

A naive strategy is to first generate a set of augmented partitions with various dimensions, e.g., $\{k, 2k, \ldots, 10k\}$, and then select the optimal parameter by grid search. However, this approach is overlimited in three ways.

1) *Undesired Hyperparameters:* It introduces several hyperparameters, requiring repetitive parameter tuning to obtain the appropriate parameter combination, which is undesirable in unsupervised learning because ground-truth labels are difficult to access [31], [32], [33], [34].
2) *Neglecting Diverse Information:* Selecting the optimal partition from the candidate partitions ignores the rich and complementary information within the diverse partitions, degrading the representation capacity.
3) *Expensive Tuning Costs:* The single-view scenario requires ten-time parameter tuning. However, it induces 10^v parameter combinations in multiview scenarios, which is unacceptable in real scenarios.

To this end, we consider designing an elegant strategy to obtain the optimal kernel partition. However, directly solving the optimal partition/parameter is challenging, as there is no prior. Inspired by the popular ensemble clustering [35], [36], [37], which combines multiple clustering into better and more robust consensus clustering, we consider adopting a diverse partition/parameter fusion approach to get the optimal kernel partition. Specifically, we predefine a search region of parameter d and generate multiple candidate augmented partitions across multiple views. Then, we design an elegant fusion strategy to adaptively integrate these partitions by reweighting their importance and, finally, fuse a consensus partition. The contributions are summarized as follows.

1) This article investigates the challenging problem of determining the optimal partition/parameter in unsupervised late fusion MKC. We propose to transform the original complex problem into a partition/parameter fusion problem, which can be easily solved.
2) We design an elegant, tuning-free, and flexible fusion mechanism called TFMKC to fuse diverse and complementary information by reweighting the importance of diverse augmented partitions across multiple views. In this way, view-level information and partition-level information are combined to learn a consensus partition. In addition, we provide an upper bound of our model.
3) Our TFMKC exhibits competitive effectiveness and efficiency over traditional fine-tuning strategies and outperforms existing baselines with large margins on eight benchmark datasets. In addition, our TFMKC achieves linear computational complexity, demonstrating promising scalability in large-scale datasets.

II. Related Work

A. Generation of Kernel Matrix

Given raw data $\{\mathbf{x}_j\}_{j=1}^n$ drawn from feature space $\mathcal{X} \in \mathbb{R}^{d_{\mathcal{X}}}$, a kernel method can encode them into a high-dimensional RKHS $\mathcal{H} \in \mathbb{R}^{d_{\mathcal{H}}}$ [38] through a nonlinear kernel mapping $\phi(\cdot)$. However, the dimension of $\mathcal{H}$ could be infinite, making it difficult to explicitly define $\phi(\cdot)$ and generate embeddings. Fortunately, Mercer's theorem [39] pointed out that we can directly calculate the inner product of mapped vectors in $\mathcal{H}$ by kernel function $\kappa(\cdot, \cdot)$ rather than defining the mapping functions $\phi(\cdot)$, i.e.,

$$\mathbf{K}_{jl} = \kappa\left(\mathbf{x}_j, \mathbf{x}_l\right) = \phi\left(\mathbf{x}_j\right)^{\top}\phi(\mathbf{x}_l). \tag{1}$$

The typical kernel generation method is to explicitly define kernel functions, such as linear, polynomial, Gaussian, Laplacian, Cauchy, and Sigmoid functions. In addition, researchers have developed specific kernel generation methods tailored to the characteristics of problems in different domains. For example, Nyström [40] is a pioneering research that selects a small subset to approximate the original kernel matrix. Learning kernel functions is another pipeline; [41] first categorized over 125 000 different types of kernel functions and parameters and then introduced sparse representation and orthogonal matching pursuit algorithm to search for the appropriate kernel functions and parameters. Recently, owing to the powerful capacity of deep learning, introducing a neural network to generate kernel matrix is a promising research; DMMV [42] utilized a neural network to optimize both the global kernel and the local view-specific kernels, exploring complementarity and diversity.

B. Multiple Kernel k-Means

k-means [43] minimizes the intercluster loss, which can be expressed by

$$\begin{aligned}&\min_{\mathbf{Y}} \sum_{j=1}^{n}\sum_{q=1}^{k} \|\mathbf{x}_j - \mathbf{c}_q\|_2^2 \mathbf{Y}_{jq}\\ &\text{s.t.} \sum_{q=1}^{k} \mathbf{Y}_{jq} = 1\end{aligned} \tag{2}$$

where $\{\mathbf{c}_q\}_{q=1}^{k}$ denote the centroids and $\mathbf{Y} \in \{0,1\}^{n\times k}$ represents the discrete indicator matrix.

According to [18], the relaxed kernel k-means (KKM) is formulated by

$$\begin{aligned}&\min_{\mathbf{H}} \operatorname{Tr}\big(\mathbf{K}\big(\mathbf{I} - \mathbf{H}\mathbf{H}^\top\big)\big)\\ &\text{s.t. } \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k\end{aligned} \tag{3}$$

where $\mathbf{H}$ is the partition matrix obtained by truncated-k EVD, and we further perform k-means to compute the predicted labels.

For multikernel settings, the combined kernel matrix is commonly supposed to be linear combinations of υ base kernels. Multiple kernel k-means (MKKM) [44] is given as

$$\begin{aligned}&\min_{\mathbf{H},\boldsymbol{\gamma}} \operatorname{Tr}\big(\mathbf{K}_{\boldsymbol{\gamma}}\big(\mathbf{I} - \mathbf{H}\mathbf{H}^\top\big)\big)\\ &\text{s.t.} \begin{cases}\mathbf{H}^\top\mathbf{H} = \mathbf{I}_k\\ \boldsymbol{\gamma}^\top\boldsymbol{\gamma} = 1, \quad \boldsymbol{\gamma} \geq \mathbf{0}\end{cases}\end{aligned} \tag{4}$$

where $\mathbf{K}_{\boldsymbol{\gamma}} = \sum_{p=1}^{\upsilon} \gamma_p^2 \mathbf{K}_p$ denotes the combined kernel and $\boldsymbol{\gamma}$ denotes the view contribution.

C. Late Fusion Multiple Kernel k-Means

Unlike the kernel fusion method that integrates a fused kernel across multiple base partitions, the late fusion paradigm aims to fuse a consensus partition across multiple base partitions $\{\mathbf{H}_p\}_{p=1}^{\upsilon}$ [24], which is formulated by

$$\begin{aligned}&\max_{\mathbf{H},\mathbf{W}_p,\boldsymbol{\gamma}} \operatorname{Tr}\big(\mathbf{H}^\top\mathbf{H}_{\boldsymbol{\gamma}} + \lambda\mathbf{H}^\top\mathbf{Q}\big)\\ &\text{s.t.} \begin{cases}\mathbf{H}^\top\mathbf{H} = \mathbf{I}_k\\ \mathbf{V}_p^\top\mathbf{V}_p = \mathbf{I}_k\\ \boldsymbol{\gamma}^\top\boldsymbol{\gamma} = 1, \quad \boldsymbol{\gamma} \geq \mathbf{0}\end{cases}\end{aligned} \tag{5}$$

where $\mathbf{H}$ represents the consistent partition, $\mathbf{H}_{\boldsymbol{\gamma}} = \sum_{p=1}^{\upsilon} \gamma_p \mathbf{H}_p\mathbf{V}_p$ denotes the sum of base partitions that are aligned by column permutation matrices, $\mathbf{V}_p$ denotes the pth permutation matrix with orthogonal constraint, $\mathbf{Q}$ is the partition from the average kernel, and λ denotes a balanced hyperparameter. Specifically, $\mathbf{Q} = \arg\max_{\mathbf{Q}^\top\mathbf{Q}=\mathbf{I}_k} \operatorname{Tr}(\mathbf{Q}^\top(\sum_{p=1}^{\upsilon}(1/\upsilon)\mathbf{K}_p)\mathbf{Q})$.

LFMKC focuses on maximizing the alignment of fused base partitions. Although it has achieved promising performance, it encounters a limited feature presentation capability problem because base partitions $\mathbf{H}_p$ are computed by truncated-k EVD, ignoring potential benefits within augmented partitions $\mathbf{U}_p$ from truncated-d $(d > k)$ EVD.

III. Methodology

A. Motivation

Recall our analysis of the limited representation capacity concern in existing late fusion MKC, i.e., precomputed partitions are fixed by truncated-k EVD, which ignores the potential benefits across diverse partitions. A naive idea to alleviate this problem is to fine-tune the size of the kernel partition. Specifically, we first generate a set of candidate augmented partitions $\mathbf{U}^{(i)} \in \mathbb{R}^{n\times d_i}$ by truncated-d_i $(d_i > k)$ EVD, e.g., $\{d_1, d_2, d_3, \ldots, d_m\}$. Then, the optimal d with satisfactory performance can be selected by grid search via maximally the alignment of the original kernel and the refined kernel $\mathbf{U}^{(i)}\mathbf{U}^{(i)\top}$, i.e.,

$$\begin{aligned}\mathbf{U}^{(i)} = \arg\max_{\mathbf{U}^{(i)}} \operatorname{Tr}\big(\mathbf{K}\mathbf{U}^{(i)}\mathbf{U}^{(i)\top}\big)\\ \text{s.t. } \mathbf{U}^{(i)\top}\mathbf{U}^{(i)} = \mathbf{I}_{d_i}.\end{aligned} \tag{6}$$

Tuning the optimal parameter is an inflexible way that incurs: 1) dataset-related results: the optimal d_{opt} varies with different datasets; 2) repetitive tuning costs: parameter tuning induces m combinations in single-kernel scenarios, while in multikernel scenarios, it induces unacceptable m^{υ} combinations; and 3) neglecting rich information across diverse partitions.

Moreover, learning the optimal parameter directly through optimization can be a complex problem, as there is no prior available in unsupervised learning. Therefore, a significant problem arises: is there a flexible strategy that can replace parameter tuning?

B. Proposed TFMKC Model

To optimize the above formulation, we consider transforming the original problem that directly determines the optimal partition/parameter into a partition/parameter fusion problem, which is inspired by ensemble clustering that fuses multiclustering into superior and robust consensus clustering.

To this end, we propose jointly utilizing the partitionwise diversity and complementary information, reweighting their contributions automatically through optimization, to displace manual selection. Taking the single-view scenario as an example, we first define the weights β_i of the augmented partitions $\mathbf{U}^{(i)}$ as follows:

$$\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_m)^\top, \quad \text{s.t. } \boldsymbol{\beta}^\top\mathbf{1}_m = 1, \ \boldsymbol{\beta} \geq \mathbf{0}. \tag{7}$$

Similar to (6), we introduce a function that can evaluate the alignment of the refined kernel $\mathbf{K}^{(i)}$ and our pursuit optimal partition, i.e.,

$$f_{\text{eval}}\big(\mathbf{K}^{(i)}, \mathbf{H}\big) = \operatorname{Tr}\big(\mathbf{K}^{(i)}\mathbf{H}\mathbf{H}^\top\big) \tag{8}$$

where $\mathbf{K}^{(i)} = \mathbf{U}^{(i)}\mathbf{U}^{(i)\top}$ denotes the refined kernel with dimension d_i and $\mathbf{H} \in \mathbb{R}^{n\times k}$ denotes the optimal consensus partition. A larger f_{eval} represents a better performance of $\mathbf{K}^{(i)}$.

Intuitively, a larger f_{eval} means a larger contribution. Given m candidate dimensions d, we can derive a weighted objective

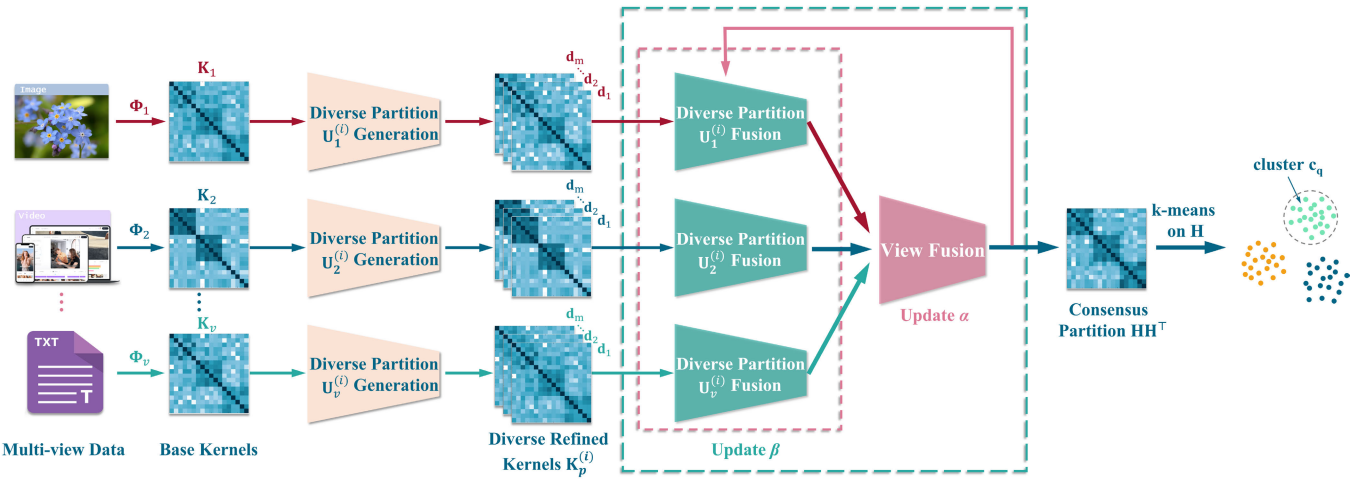


Fig. 2. Framework of our TFMKC method. The core idea is to learn a consensus partition $\mathbf{H}$ by jointly fusing partition- and view-level diverse and complementary information. Specifically, TFMKC first performs KKM to generate diverse augmented partitions $\mathbf{U}_p^{(i)}$; note that $\mathbf{U}_p^{(i)}\mathbf{U}_p^{(i)\top}$ can recreate the original structures of the pth base kernel $\mathbf{K}_p$ and then fuses these augmented base partitions $\mathbf{U}_p^{(i)}$ by reweighting importance, which is different from traditional fine-tuning method. In this way, rich feature representation across multiple views can be integrated. The subsequent process is to perform k-means on $\mathbf{H}$ to get the discrete clustering labels.

by maximizing a convex combination of a set of evaluation functions, i.e.,

$$\begin{aligned}\boldsymbol{\beta} &= \arg\max_{\boldsymbol{\beta}} \sum_{i=1}^{m} \beta_i \mathrm{Tr}\big(\mathbf{K}^{(i)}\mathbf{H}\mathbf{H}^\top\big) \\ &= \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{m} (1-\beta_i)\mathrm{Tr}\big(\mathbf{K}^{(i)}\mathbf{H}\mathbf{H}^\top\big) \\ &\text{s.t. } \boldsymbol{\beta}^\top \mathbf{1}_m = 1, \quad \boldsymbol{\beta} \geq \mathbf{0}. \end{aligned} \tag{9}$$

Equation (9) provides an intuitive solution to measure the importance of partitions. However, it is not a viable strategy since it incurs a sparse trivial solution; namely, the weight of the largest f_{eval} will equal one, while the others are 0. This is equivalent to manual selection and reporting the best result. Such a method is called TFMKC-tune.

To alleviate this problem, we modify (9) by solving quadratic programming (QP) of $\boldsymbol{\beta}$ as follows:

$$\begin{aligned}\boldsymbol{\beta} &= \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{m} (1-\beta_i)^2 \mathrm{Tr}\big(\mathbf{K}^{(i)}\mathbf{H}\mathbf{H}^\top\big) \\ &= \arg\max_{\boldsymbol{\beta}} \sum_{i=1}^{m} \left(\beta_i - \frac{1}{2}\beta_i^2\right) \mathrm{Tr}\big(\mathbf{K}^{(i)}\mathbf{H}\mathbf{H}^\top\big) \\ &\text{s.t. } \boldsymbol{\beta}^\top \mathbf{1} = 1, \quad \boldsymbol{\beta} \geq \mathbf{0}. \end{aligned} \tag{10}$$

Compared to TFMKC-tune with a trivial solution, the weight distribution of our TFMKC is smoother. In particular, introducing fusion mechanisms of diverse augmented partitions into optimization can circumvent the repetitive fine-tuning achieved by assigning different weights according to their importance. That is, expressive augmented partitions are imposed large weights, while the limited ones are assigned small weights.

In the MKC scenario, it is generally assumed that all views are helpful. To integrate diverse and complementary information across multiple kernels, we extend (10) to an MKC version, and our TFMKC is given as follows:

$$\begin{aligned}&\max_{\mathbf{H},\boldsymbol{\omega},\boldsymbol{\beta}} \sum_{p=1}^{v} \omega_p \sum_{i=1}^{m} \left(\beta_{ip} - \frac{1}{2}\beta_{ip}^2\right) \mathrm{Tr}\big(\mathbf{K}_p^{(i)}\mathbf{H}\mathbf{H}^\top\big) \\ &\text{s.t. } \begin{cases} \boldsymbol{\omega}^\top \boldsymbol{\omega} = 1, & \boldsymbol{\omega} \geq \mathbf{0} \\ \boldsymbol{\beta}^\top \mathbf{1}_m = \mathbf{1}_v, & \boldsymbol{\beta} \geq \mathbf{0} \\ \mathbf{H}^\top \mathbf{H} = \mathbf{I}_k \end{cases} \end{aligned} \tag{11}$$

where $\boldsymbol{\omega}_p$ represents the view contribution. Since imposing ℓ_1-norm constraint $\boldsymbol{\omega}^\top \mathbf{1}_v = 1$ will induce a trivial solution that MKC is degraded into single-kernel clustering, we impose ℓ_2-norm constraint on kernel weights by following [24]. Note that $\boldsymbol{\beta} \in \mathbb{R}^{m \times v}$ represents the weight matrix of augmented partitions $\mathbf{U}_p^{(i)}$. Therefore, our tuning-free model integrates view-weight $\boldsymbol{\omega}$ and partition-weight $\boldsymbol{\beta}$ into a unified framework.

Remark 1: The proposed TFMKC requires initializing a search region of dimension d_i as input, e.g., feature dimensions can vary in $\{d_1, d_2, \ldots, d_m\}$. In fact, for late fusion MKC, we only need to compute the largest augmented partition $\mathbf{U}_p^{(m)} \in \mathbb{R}^{n \times d_m}$, and other smaller partitions can be extracted from the largest one, which significantly reduces computation.

Remark 2: Note that we only initialize a search region of feature dimension d, and no extra hyperparameters are involved in optimization. Therefore, our TFMKC can be regarded as a parameter-free model, which is practical in unsupervised learning.

In summary, we first transform the challenging task of directly determining the optimal partition/parameter into a flexible partition/parameter ensemble problem and propose a tuning-free fusion mechanism to integrate partition level and view level by adaptively reweighting their importance, distinguishing our work from existing parameter-tuning methods. Fig. 2 illustrates the framework of our TFMKC method.

C. Solver

Considering the nonconvexity of (11), this section designs a three-step block-coordinate descent optimization with each step having a closed-form solution.

1) Update $\boldsymbol{H}$: With $\boldsymbol{\beta}$ and $\boldsymbol{\omega}$ been fixed, (11) is converted to

$$\max_{\mathbf{H}} \operatorname{Tr}\left(\mathbf{K}_{\text{inte}}\mathbf{H}\mathbf{H}^{\top}\right), \quad \text{s.t. } \mathbf{H}^{\top}\mathbf{H} = \mathbf{I}_k \tag{12}$$

where $\mathbf{K}_{\text{inte}} = \sum_{p=1}^{\upsilon} \omega_p \sum_{i=1}^{m} (\beta_{ip} - (1/2)\beta_{ip}^2)\mathbf{K}_p^{(i)}$.

Theorem 1 illustrates that the optimal $\mathbf{H}$ can be obtained by performing truncated-k SVD on $\mathbf{B} \in \mathbb{R}^{n\times \upsilon d_m}$ rather than using EVD on $\mathbf{K}_{\text{inte}} \in \mathbb{R}^{n\times n}$, which significantly reduces computational complexity.

Theorem 1: Given $\mathbf{K}_p^{(i)}$ defined in (8), the optimal $\mathbf{H}$ obtained by truncated-k EVD of $\mathbf{K}_{\text{inte}}$ is equivalent to the solution solved by the truncated-k SVD of $\mathbf{B} = (\mathbf{B}_1, \mathbf{B}_2, \ldots, \mathbf{B}_{\upsilon})$, $\mathbf{B} \in \mathbb{R}^{n\times \upsilon d_m}$. Specifically,

$$\mathbf{B}_p = \mathbf{U}_p^{(m)} \operatorname{diag}\left(\sqrt{\mathbf{D}\mathbf{1}_{m\times m}^{(\mathrm{u})}\boldsymbol{\Gamma}_{:,p}}\right), \quad p \in \{1, 2, \ldots, \upsilon\} \tag{13}$$

where $\mathbf{D}$ is a diagonal block matrix with $\mathbf{1}_{d_i - d_{i-1}}$ as the ith block, $\mathbf{1}_{d_i-d_{i-1}} = (1, 1, \ldots, 1)^{\top}$ denotes a $(d_i - d_{i-1})$-dimensional column vector with all elements being 1, $\mathbf{1}_{m\times m}^{(\mathrm{u})}$ denotes the upper triangular part of the $m \times m$ all-one matrix $\mathbf{1}_{m\times m}$, and $\boldsymbol{\Gamma}_{ip} = (\beta_{ip} - (1/2)\beta_{ip}^2)\omega_p$. Mathematically, $\mathbf{D}$ is defined by

$$\mathbf{D} = \operatorname{blkdiag}\left(\mathbf{1}_{d_1}, \mathbf{1}_{d_2-d_1}, \ldots, \mathbf{1}_{d_m-d_{m-1}}\right) \tag{14}$$

where blkdiag$(\cdot)$ is a function used to create a block diagonal matrix with input arguments. Specifically, with multiple input arguments that represent a matrix, it places these matrices along the diagonal with zeros filling the remaining positions.

Proof: This can be verified by checking $\mathbf{K}_{\text{inte}} = \mathbf{B}\mathbf{B}^{\top}$. According to (6), we have $\mathbf{K}_p^{(i)} = \mathbf{U}_p^{(i)}\mathbf{U}_p^{(i)\top}$ and $\mathbf{U}_p^{(i)} = \mathbf{U}_p^{(m)}\begin{bmatrix} \mathbf{I}_{d_i} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$. Then, we have

$$\mathbf{K}_p^{(i)} = \mathbf{U}_p^{(m)}\begin{bmatrix} \mathbf{I}_{d_i} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}\mathbf{U}_p^{(m)\top}. \tag{15}$$

Naturally, we have

$$\begin{aligned} &\omega_p \sum_{i=1}^{m}\left(\beta_{ip} - \frac{1}{2}\beta_{ip}^2\right)\mathbf{K}_p^{(i)} \\ &= \mathbf{U}_p^{(m)} \sum_{i=1}^{m} \omega_p \left(\beta_{ip} - \frac{1}{2}\beta_{ip}^2\right)\begin{bmatrix} \mathbf{I}_{d_i} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}\mathbf{U}_p^{(m)\top}. \end{aligned} \tag{16}$$

Furthermore, we have

$$\begin{aligned} &\sum_{i=1}^{m} \omega_p \left(\beta_{ip} - \frac{1}{2}\beta_{ip}^2\right)\begin{bmatrix} \mathbf{I}_{d_i} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \\ &= \sum_{i=1}^{m} \boldsymbol{\Gamma}_{ip}\begin{bmatrix} \mathbf{I}_{d_i} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{I}_{d_1} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{I}_{d_m-d_{m-1}} \end{bmatrix} \operatorname{diag}\left(\mathbf{D}\mathbf{1}_{m\times m}^{(\mathrm{u})}\boldsymbol{\Gamma}_{:,p}\right) \\ &= \operatorname{diag}\left(\mathbf{D}\mathbf{1}_{m\times m}^{(\mathrm{u})}\boldsymbol{\Gamma}_{:,p}\right). \end{aligned} \tag{17}$$

When matrix $\mathbf{D}$ is left-multiplied onto another matrix, it duplicates the rows of the latter, effectively transforming a matrix with m rows into one with d_m rows. Consequently, we derive that

$$\begin{aligned} \mathbf{K}_{\text{inte}} &= \sum_{p=1}^{\upsilon} \mathbf{U}_p^{(m)} \operatorname{diag}\left(\mathbf{D}\mathbf{1}_{m\times m}^{(\mathrm{u})}\boldsymbol{\Gamma}_{:,p}\right)\mathbf{U}_p^{(m)\top} \\ &= \sum_{p=1}^{\upsilon} \mathbf{B}_p\mathbf{B}_p^{\top} = \mathbf{B}\mathbf{B}^{\top} \end{aligned} \tag{18}$$

which indicates that the optimal $\mathbf{H}$ can be obtained by performing truncated-k SVD on $\mathbf{B} \in \mathbb{R}^{n\times \upsilon d_m}$ rather than using EVD on $\mathbf{K}_{\text{inte}} \in \mathbb{R}^{n\times n}$, which significantly reduces computational complexity.

This completes the proof. □

2) Update $\boldsymbol{\beta}$: With $\mathbf{H}$ and $\boldsymbol{\omega}$ been fixed, (11) can be independently solved across υ kernels, and each column of $\boldsymbol{\beta}$ is a QP problem, i.e.,

$$\begin{aligned} &\min_{\boldsymbol{\beta}_{:,p}} \frac{1}{2}\boldsymbol{\beta}_{:,p}^{\top}\mathbf{S}_p\boldsymbol{\beta}_{:,p} + \mathbf{e}_p^{\top}\boldsymbol{\beta}_{:,p} \\ &\text{s.t. } \boldsymbol{\beta}_{:,p}^{\top}\mathbf{1}_m = 1, \boldsymbol{\beta}_{:,p} \geq \mathbf{0} \end{aligned} \tag{19}$$

where $\mathbf{S}_p$ is a diagonal matrix, $(\mathbf{S}_p)_{ii} = \operatorname{Tr}(\mathbf{K}_p^{(i)}\mathbf{H}\mathbf{H}^{\top})$, and $e_{pi} = -\operatorname{Tr}(\mathbf{K}_p^{(i)}\mathbf{H}\mathbf{H}^{\top})$.

Theorem 2 shows that (19) has closed solution.

Theorem 2: Define the following function with respect to x:

$$g_p(x) := \left(\max\left(\frac{x\mathbf{1}_m - \mathbf{e}_p}{\operatorname{diag}(\mathbf{S}_p)}, 0\right)\right)^{\top}\mathbf{1}_m - 1. \tag{20}$$

The closed-form solution (19) is

$$\boldsymbol{\beta}_{:,p} = \max\left(\frac{\mu_p\mathbf{1}_m - \mathbf{e}_p}{\operatorname{diag}(\mathbf{S}_p)}, 0\right) \tag{21}$$

where μ_p is the root of function $g_p(x)$.

Proof: For (19), the Lagrangian function of $\boldsymbol{\beta}_{:,p}$ is

$$\begin{aligned} &\mathcal{L}\left(\boldsymbol{\beta}_{:,p}, \mu_p, \boldsymbol{\nu}_p\right) \\ &= \frac{1}{2}\boldsymbol{\beta}_{:,p}^{\top}\mathbf{S}_p\boldsymbol{\beta}_{:,p} + \mathbf{e}_p^{\top}\boldsymbol{\beta}_{:,p} - \mu_p\left(\boldsymbol{\beta}_{:,p}^{\top}\mathbf{1}_m - 1\right) - \boldsymbol{\nu}_p^{\top}\boldsymbol{\beta}_{:,p} \end{aligned} \tag{22}$$

where scalar μ_p and vector $\boldsymbol{\nu}_p \geq \mathbf{0}$ denote the Lagrangian multipliers. We derive the derivative of (22) with respect to $\boldsymbol{\beta}_{:,p}$ and enforce it equal zero; we obtain

$$\mathbf{S}_p\boldsymbol{\beta}_{:,p} + \mathbf{e}_p - \mu_p\mathbf{1}_m - \boldsymbol{\nu}_p = \mathbf{0}. \tag{23}$$

Algorithm 1 TFMKC

Input: Augmented partitions $\{\mathbf{U}_p^{(i)}\}_{p=1}^{\upsilon}$ and k.
Output: The optimal $\mathbf{H}$, $\boldsymbol{\omega}$ and $\boldsymbol{\beta}$.

```
Initialize ω_p = 1/υ, β_ip = 1/m.
while not converge do
    Update H by optimizing Eq. (12),
    Update β by optimizing Eq. (19),
    Update ω by optimizing Eq. (26),
end while
Utilize k-means on H to discrete the clustering labels.
```

According to the KKT condition, we have $\boldsymbol{\beta}_{:,p}^{\top}\boldsymbol{v}_p = 0$. Therefore, we derive

$$\boldsymbol{\beta}_{:,p} = \max\left(\frac{\mu_p \mathbf{1}_m - \mathbf{e}_p}{\operatorname{diag}(\mathbf{S}_p)}, 0\right). \tag{24}$$

Then, according the constraint $\boldsymbol{\beta}_{:,p}^{\top}\mathbf{1}_m = 1$, we have the following equation:

$$g_p(\mu_p) = 0. \tag{25}$$

It is clear that μ_p is the root of the function $g_p(x)$, which is piecewise linear and increases monotonically. The root can be solved by Newton's method efficiently.

This completes the proof. □

3) Update $\boldsymbol{\omega}$: With $\mathbf{H}$ and $\boldsymbol{\beta}$ been fixed, the proposed algorithm is converted to

$$\begin{aligned} &\max_{\boldsymbol{\omega}} \sum_{p=1}^{v} \omega_p \tau_p \\ &\text{s.t. } \boldsymbol{\omega}^{\top}\boldsymbol{\omega} = 1, \boldsymbol{\omega} \geq \mathbf{0} \end{aligned} \tag{26}$$

where $\tau_p = \sum_{i=1}^{m}(\beta_{ip} - (1/2)\beta_{ip}^2)\operatorname{Tr}(\mathbf{K}_p^{(i)}\mathbf{H}\mathbf{H}^{\top})$. This problem can be solved by calculating $\boldsymbol{\omega} = \boldsymbol{\tau}/\|\boldsymbol{\tau}\|_2$.

D. Analysis and Extensions

1) *Computational Complexity:* The workflow of Algorithm 1 consists of three components. First, updating $\mathbf{H}$ involves performing SVD, incurring a cost of $\mathcal{O}(nv^2d_m^2)$. Second, updating $\boldsymbol{\beta}$ requires $\mathcal{O}(mv)$ operations to obtain the closed-form solution. Finally, updating $\boldsymbol{\omega}$ necessitates $\mathcal{O}(mv)$ operations. Thus, the time complexity of optimization is $\mathcal{O}(nv^2d_m^2)$.
Note that our model requires k-means to output discrete labels, which needs time complexity $\mathcal{O}(nk^2)$. Thus, the overall computational complexity is $\mathcal{O}(nv^2d_m^2)$, which is linear concerning sample number, making it can be scaled to large-scale tasks.

2) *Convergence:* The three-step alternative optimization involves one SVD and two convex problems. Since each suboptimization has a closed-form solution, the objective of Algorithm 1 increases monotonically with the iteration. Moreover, Theorem 3 illustrates that it is upper bounded by $(1 - (1/2m))\sqrt{v}k$. Therefore, our TFMKC theoretically converges to a local maximum, as pointed out in [45].
Theorem 3: An upper bound of (11) is $(1 - (1/2m))\sqrt{v}k$.
Proof: According to constraints $\mathbf{U}^{(i)\top}\mathbf{U}^{(i)} = \mathbf{I}_{d_i}$, $\mathbf{K}_p^{(i)} = \mathbf{U}^{(i)}\mathbf{U}^{(i)\top}$, and $\mathbf{H}^{\top}\mathbf{H} = \mathbf{I}_k$, we have $\operatorname{Tr}(\mathbf{K}_p^{(i)}\mathbf{H}\mathbf{H}^{\top}) \leq k$, i.e., $f_{\text{eval}}(\mathbf{K}_p^{(i)}, \mathbf{H}) \leq k$. Therefore, we derive that

$$\begin{aligned} &\max_{\mathbf{H},\boldsymbol{\omega},\boldsymbol{\beta}} \sum_{p=1}^{v} \omega_p \sum_{i=1}^{m}\left(\beta_{ip} - \frac{1}{2}\beta_{ip}^2\right)\operatorname{Tr}\left(\mathbf{K}_p^{(i)}\mathbf{H}\mathbf{H}^{\top}\right) \\ &\leq \max_{\boldsymbol{\omega},\boldsymbol{\beta}} \sum_{p=1}^{v} \omega_p \sum_{i=1}^{m}\left(\beta_{ip} - \frac{1}{2}\beta_{ip}^2\right)k \\ &= \sum_{p=1}^{v} \frac{1}{\sqrt{v}} \sum_{i=1}^{m}\left(\frac{1}{m} - \frac{1}{2m^2}\right)k \\ &= \left(1 - \frac{1}{2m}\right)\sqrt{v}k. \end{aligned} \tag{27}$$

Thus, (11) is upper bounded by $(1 - (1/2m))\sqrt{v}k$. This completes the proof. □

3) *Extensions:* The proposed TFMKC provides an elegant strategy to address the limited representation capacity problem in the late fusion MKC community. Most of all, our tuning-free strategy will avoid undesired parameter tuning, significantly improving efficiency. Moreover, such a parameter fusion strategy shows strong scalability that can be easily extended to existing MVC methods to exploit potential information.

TABLE I
BENCHMARK MKC DATASETS

Datasets	Scenario	Instances	Views	Clusters
Politicsuk	Graph	419	3	5
Rugby	Graph	854	3	15
Willow	Image	911	3	7
Plant	Image	940	69	4
Flower17	Image	1,360	7	17
CCV	Video	6,773	3	20
Flower102	Image	8,189	4	102
Reuters	Text	18,758	5	6

IV. EXPERIMENT

A. Synthetic Dataset

In this article, we design a synthetic dataset to show our motivation. Note that the synthetic dataset serves two purposes: 1) to visualize the limited representation issue in existing late fusion MKC methods and 2) the difficulty in determining the optimal feature dimension.

Specifically, we first generate raw features $\mathbf{X}_{\text{syn}}$ by random sampling from three different normal distributions, each cluster contains six samples, and each instance has nine dimensions. Then, we construct a linear kernel matrix $\mathbf{K}_{\text{syn}} = \mathbf{X}_{\text{syn}}^{\top}\mathbf{X}_{\text{syn}}$. The results in Fig. 1 are based on $\mathbf{K}_{\text{syn}}$.

B. Real-World Datasets

Table I lists eight public MKC datasets with different types and sizes, including Politicsuk,[1] Rugby,[2] Willow,[3] Plant,[4] Flower17,[5] CCV,[6] Flower102,[7] and Reuters.[8] These datasets can be downloaded from public websites.

[1] http://mlg.ucd.ie/aggregation/
[2] http://mlg.ucd.ie/aggregation/
[3] https://www.di.ens.fr/willow/research/stillactions/
[4] https://bmi.inf.ethz.ch/supplements/protsubloc
[5] http://www.robots.ox.ac.uk/~vgg/data/flowers/
[6] https://www.ee.columbia.edu/ln/dvmm/CCV/
[7] http://www.robots.ox.ac.uk/~vgg/data/flowers/
[8] http://kdd.ics.uci.edu/databases/reuters21578.html

TABLE II

COMPARING THE CLUSTERING METRICS. THE STRONGEST METRICS ARE SYMBOLIZED IN BOLD, WHILE THE SUBOPTIMAL RESULTS ARE ITALICIZED AND UNDERLINED. "-" REPRESENTS TIME-OUT OR OUT-OF-MEMORY ERRORS. "RANK" REPORTS THE AVERAGE RANKING PERFORMANCE

Datasets	AKKM	MKKM	MKKM-MR	SwMC	ONKC	LFMKC	SPMKC	CAGL	OPLF	Proposed
Number of parameter	0	0	1	0	2	1	2	2	0	0
	NMI (%)									
Politicsuk	39.95	40.07	42.00	9.80	41.29	*42.14*	32.04	24.65	32.72	**44.91**
Rugby	36.93	36.45	*48.78*	6.62	48.02	43.24	45.22	41.22	44.55	**52.77**
Willow	5.70	5.70	6.31	1.13	6.07	7.91	8.14	4.78	*8.61*	**10.99**
Plant	*26.53*	19.49	20.37	0.51	10.49	23.35	0.21	11.90	13.67	**30.55**
Flower17	49.70	44.89	56.36	2.38	52.65	*58.90*	38.71	57.18	51.51	**61.28**
CCV	16.84	15.04	18.03	1.07	18.52	*20.09*	1.60	6.08	18.72	**23.99**
Flower102	46.02	42.67	*56.71*	5.51	56.11	54.94	-	45.09	46.77	**61.81**
Reuters	27.37	27.35	25.30	1.35	22.27	*27.39*	-	-	27.09	**34.54**
Rank	5.63	6.75	3.88	9.50	5.00	3.13	7.17	7.14	4.88	1.00
	ACC (%)									
Politicsuk	56.78	57.12	*60.69*	47.97	60.04	59.86	51.55	50.84	49.88	**72.36**
Rugby	42.48	41.96	*55.58*	24.59	55.38	47.52	49.41	54.33	51.87	**58.14**
Willow	22.19	22.03	22.90	19.76	22.60	26.38	*27.77*	22.72	25.58	**32.30**
Plant	*61.28*	56.05	50.27	38.94	41.43	59.53	32.87	43.09	48.51	**61.76**
Flower17	50.83	44.87	58.51	7.06	54.17	*61.02*	35.51	50.51	50.66	**62.58**
CCV	19.63	17.99	21.24	10.84	22.39	*25.13*	9.67	12.58	22.74	**27.37**
Flower102	27.07	22.41	*40.22*	6.72	39.55	38.45	-	26.25	29.78	**48.48**
Reuters	45.46	45.44	*46.15*	27.12	41.85	45.68	-	-	44.65	**48.00**
Rank	5.63	6.88	3.25	9.38	4.88	3.50	7.33	6.71	5.50	1.00
	Purity (%)									
Politicsuk	78.50	78.70	**81.11**	48.45	80.62	78.94	79.00	56.32	72.55	*81.09*
Rugby	60.36	60.18	*72.70*	29.16	72.07	66.14	68.15	55.27	67.21	**75.83**
Willow	27.10	27.15	27.48	19.98	27.30	29.89	*29.97*	23.27	29.31	**33.52**
Plant	*61.28*	56.05	56.71	39.36	49.02	59.53	39.15	46.81	52.87	**62.91**
Flower17	51.95	46.24	59.66	8.24	55.40	*62.40*	37.79	52.50	52.28	**63.98**
CCV	23.75	22.18	23.74	11.35	24.64	*28.16*	11.78	13.58	26.52	**31.29**
Flower102	32.27	27.79	*46.34*	8.08	45.63	44.56	-	31.33	34.03	**55.01**
Reuters	53.01	52.94	52.15	28.25	52.63	*53.23*	-	-	52.92	**56.49**
Rank	5.63	6.63	3.75	9.50	4.50	3.38	6.33	7.86	5.25	1.13
	ARI (%)									
Politicsuk	37.95	38.36	*42.17*	4.27	39.67	40.93	31.54	14.25	25.98	**53.99**
Rugby	22.55	21.99	*36.85*	-0.84	35.97	27.21	28.72	22.95	31.21	**38.86**
Willow	3.11	3.20	3.44	0.03	3.29	4.56	*5.57*	0.28	4.62	**7.53**
Plant	*24.64*	17.42	19.03	-0.25	9.78	21.66	-0.39	1.76	12.28	**28.57**
Flower17	32.16	27.25	39.87	0.02	35.23	*44.10*	19.58	30.48	35.03	**46.56**
CCV	6.60	5.75	7.15	-0.02	7.74	*9.44*	-0.01	0.29	8.02	**10.89**
Flower102	15.46	12.06	*25.49*	0.10	24.86	25.46	-	2.28	18.92	**35.03**
Reuters	21.84	21.82	*23.08*	0.09	20.32	22.10	-	-	21.25	**26.13**
Rank	5.75	6.63	3.13	9.50	4.88	3.25	7.00	8.00	5.00	1.00

C. Compared Baselines

Nine existing kernel and graph MVC models are set as baselines.

1) AKKM performs KKM on the average kernel.
2) MKKM [44] is a pioneering MKC method that aligns the fused kernel and the consensus partition.
3) MKKM-MR [46] induces a matrix-induced regularizer to measure the view similarity, serving to improve the diversity of kernels.
4) SwMC [47] designs an autoweighted strategy to measure the contribution of kernels, without explicitly introducing view weights.
5) ONKC [48] claims to learn the optimal neighborhood kernel of input kernels.
6) LFMKC [24] is a pioneering late-fusion-based kernel clustering method that aims to maximize the alignment between kernel partitions and the consensus optimal one.
7) SPMKC [13] jointly explores the local and global structures across multiple kernels and introduces a similarity graph to explore the neighbor information.
8) CAGL [49] introduces a Laplacian rank constraint into graph learning to explore a consensus similarity graph with clear diagonal block structures.
9) OPLF [27] is a one-stage late-fusion method that can directly generate clustering labels without postprocessing.

TABLE III
COMPARING THE CLUSTERING METRICS ON FOUR RECENT BASELINES

Datasets	FAMKKM	SimpleMKKM	LSWMKC	MPS	Proposed	FAMKKM	SimpleMKKM	LSWMKC	MPS	Proposed
Hyper-parameters	2	0	1	2	0	2	0	1	2	0
	NMI (%)					ACC (%)				
Politicsuk	39.67	35.32	32.35	**45.12**	*44.91*	*62.02*	55.17	54.18	60.42	**72.36**
Rugby	*46.31*	41.93	34.06	45.96	**52.77**	49.83	48.21	44.03	*50.22*	**58.14**
Willow	*9.37*	6.18	6.38	8.62	**10.99**	*29.25*	22.75	24.59	27.24	**32.30**
Plant	7.06	18.85	**34.96**	30.53	*30.55*	38.11	50.57	*62.34*	**63.65**	61.76
Flower17	52.46	*55.30*	53.49	54.65	**61.28**	53.98	*54.69*	54.63	51.93	**62.58**
CCV	7.06	17.96	14.46	*22.81*	**23.99**	**38.11**	21.65	19.71	*27.38*	27.37
Flower102	56.11	57.87	51.03	*58.19*	**61.81**	41.53	41.37	32.13	*42.46*	**48.48**
Reuters	*29.90*	27.42	16.43	28.59	**34.54**	47.95	*48.64*	41.42	**48.77**	48.00
Rank	3.50	3.63	4.13	2.50	1.25	3.00	3.63	4.25	2.38	1.75
	Purity (%)					F-score (%)				
Politicsuk	78.16	76.42	78.04	**84.62**	*81.09*	40.63	34.37	28.86	*45.01*	**53.99**
Rugby	*69.15*	65.58	58.20	68.69	**75.83**	*31.57*	29.29	23.78	30.33	**38.86**
Willow	*32.79*	27.23	29.09	30.40	**33.52**	*6.02*	3.32	3.91	5.41	**7.53**
Plant	42.08	56.56	*63.94*	**64.09**	62.91	5.03	18.72	**31.26**	*29.22*	28.57
Flower17	55.13	*56.37*	55.81	55.20	**63.98**	36.00	*38.17*	37.77	36.05	**46.56**
CCV	**42.08**	24.99	23.70	30.19	*31.29*	5.03	7.42	5.65	*10.78*	**10.89**
Flower102	47.47	47.51	38.80	*49.48*	**55.01**	*28.89*	27.76	20.12	28.87	**35.03**
Reuters	*55.41*	53.43	47.12	53.94	**56.49**	25.16	24.54	6.91	*25.22*	**26.13**
Rank	3.00	3.88	4.13	2.50	1.50	3.38	3.75	4.00	2.63	1.25

10) SimpleMKKM [14] pioneers the kernel alignment criterion into MKC and derives a min–max problem with global optimum.
11) FAMKKM [28] learns the approximate partitions of the original kernel partitions and aligns a consensus partition to both original and approximate partitions.
12) LSWMKC [19] introduces graph learning into MKC to fuse a discriminative affinity graph and further learns an optimal neighborhood kernel that preserves local structures.
13) MPS [29] jointly explores the structural information within raw features and kernels by integrating multiscale partitions and enforcing the alignment between the optimal partition and local similarity graph. It also designs sparse regularization to select high-quality kernels.

The compared algorithms include: 1) kernel fusion models, namely, AKKM, MKKM, MKKM-MR, ONKC, and SimpleMKKM; 2) graph model SwMC; 3) kernel coupled graph models, i.e., SPMKC, CAGL, and LSWMKC; and 4) late fusion models, LFMKC, OPLF, FAMKKM, and MPS. The first three share the same computational complexity of $\mathcal{O}(n^3)$, and the last has a complexity of $\mathcal{O}(n)$.

D. Experimental Settings

Following existing experimental settings in clustering, the cluster number k is assumed to be known in advance, and the datasets are first centered and then normalized [17]. To mitigate the randomness of k-means in postprocessing, we initialize clustering centroids 20 times and present the average results.

The hyperparameters of the compared models are carefully tuned following the authors' settings. For our TFMKC, we initialize a search region parameter $d \in \{k, 2k, \ldots, 20k\}$, and the optimal parameter ensemble process is left for optimization. Since no extra hyperparameters are fine-tuned during optimization, our TFMKC can be considered a parameter-free model, making it suitable for unsupervised learning tasks.

We utilize four metrics, including normalized mutual information (NMI) [50], accuracy (ACC) [51], purity [52], and adjusted rand index (ARI) [53], to measure the clustering performance. Note that ARI ranges $[-1, 1]$. Compared to the rand index (RI) that ranges $[0, 1]$, ARI is a more stringent metric since it considers the randomness in results and adjusts the expected similarity under random clustering. Specifically, negative values indicate random results, 0 indicates no similarity between predicted labels and ground truth, and 1 indicates perfect agreement between predicted labels and ground truth. All the experiments are executed on a desktop with Intel i7-8700K (3.70 GHz), 64-GB RAM, and MATLAB (2020b).

E. Clustering Metrics

Tables II and III compare four clustering metrics. For reference, we provide the average rank. We find that the following holds.

1) Our designed TFMKC model achieves the strongest clustering metrics across all datasets. In particular, it outperforms the second-best ACC with large margins of 11.67%, 2.56%, 4.53%, 0.48%, 1.56%, 2.24%, 8.26%, 1.85%, 10.34%, 7.92%, 3.05%, 7.89%, and 6.02%, respectively. Our TFMKC ranks first, demonstrating its superior performance over other existing kernel or graph-based algorithms.
2) Compared to traditional late fusion MKC methods that use truncated-k EVD manner with limited representation capacity, our flexible partition fusion strategy integrates diverse and complementary information and achieves significant improvements.
3) Note that most baselines achieve positive ARI except for SwMC and SPMKC. However, the ARI metric of

TABLE IV
COMPARISON OF RUNNING TIME (S). "TUNING" MEANS THE TOTAL EXECUTION TIME. "RUNNING" MEANS THE SINGLE RUNNING TIME. "-" MEANS A HYPERPARAMETER-FREE ALGORITHM. "-" MEANS TIME-OUT OR OUT-OF-MEMORY ERRORS

Datasets	Time (s)	AKKM	MKKM	MKKM-MR	SwMC	ONKC	LFMKC	SPMKC	CAGL	OPLF	Proposed
Politicsuk	Tuning	-	-	1.63	-	144.77	0.63	54.39	314.60	1.24	-
	Running	0.01	0.03	0.05	4.34	0.15	0.02	1.51	1.94	0.06	0.12
Rugby	Tuning	-	-	7.78	-	1072.79	2.32	454.03	627.67	3.79	-
	Running	0.02	0.14	0.25	23.88	1.12	0.07	12.61	3.87	0.19	0.57
Willow	Tuning	-	-	7.48	-	572.44	1.99	131.39	1242.07	3.88	-
	Running	0.02	0.14	0.24	23.04	0.60	0.06	3.65	7.67	0.19	0.50
Plant	Tuning	-	-	1023.93	-	35119.36	27.07	1186.47	11518.35	21.40	-
	Running	0.17	3.66	33.03	116.85	36.54	0.87	32.96	71.10	1.07	4.69
Flower17	Tuning	-	-	72.28	-	5174.24	12.84	721.95	4077.47	11.71	-
	Running	0.08	1.44	2.33	72.07	5.38	0.41	20.05	25.17	0.59	4.36
CCV	Tuning	-	-	1598.44	-	347868.59	208.92	364927.26	181525.46	148.10	-
	Running	1.64	42.67	51.56	5512.67	361.99	6.74	10136.87	1120.53	7.40	76.82
Flower102	Tuning	-	-	4343.86	-	624586.09	1195.65	N/A	382874.57	973.52	-
	Running	7.27	109.76	140.12	7612.89	649.93	38.57	N/A	2363.42	48.68	489.89
Reuters	Tuning	-	-	49510.69	-	3743171.77	1571.38	N/A	N/A	983.08	-
	Running	11.43	830.46	1597.12	84059.26	3895.08	50.69	N/A	N/A	49.15	149.36

TABLE V
COMPARISON OF RUNNING TIME WITH FOUR RECENT COMPARED BASELINES

Datasets	Time(s)	FAMKKM	SimpleMKKM	LSWMKC	MPS	Proposed
Politicsuk	Tune Time	1.33	-	10.20	35.03	-
	Run Time	0.15	1.20	0.93	0.43	0.12
Rugby	Tune Time	5.94	-	57.02	177.43	-
	Run Time	0.66	4.70	5.18	2.19	0.57
Willow	Tune Time	5.10	-	76.02	162.07	-
	Run Time	0.57	3.41	6.91	2.00	0.50
Plant	Tune Time	40.74	-	478.40	5776.75	-
	Run Time	4.53	643.22	43.49	71.32	4.69
Flower17	Tune Time	25.78	-	342.02	1299.06	-
	Run Time	2.86	36.37	31.09	16.04	4.36
CCV	Tune Time	223.27	-	32625.38	62748.67	-
	Run Time	24.81	251.79	2965.94	774.67	76.82
Flower102	Tune Time	622.52	-	80613.35	121514.75	-
	Run Time	69.17	2251.63	7328.49	1500.18	489.89
Reuters	Tune Time	2516.48	-	634371.66	96717.17	-
	Run Time	279.61	2277.13	57670.15	1194.04	149.36

SwMC is close to zero on most datasets, and the ARI metric of SPMKC is negative/zero on Plant and CCV datasets, illustrating their clustering results involving obvious randomness.

4) Table III compares the clustering metrics with four recent baselines, namely, SimpleMKKM, FAMKKM, LSWMKC, and MPS. Although our NMI is inferior to them on Politicsuk and Plant datasets, our TFMKC achieves better performance on most datasets, especially on large-scale datasets, illustrating our effectiveness.

F. Running Time

Tables IV and V record the CPU time. "-" indicates no tuning process for a parameter-free algorithm, while "N/A" denotes unreported metrics caused by out-of-memory or time-out errors. As noted in Remark 2, our model can be considered a parameter-free method since we initialize the search region of parameter d and leave the rest for optimization. From the table, we see that the following holds.

1) Comparing the kernel or graph fusion baselines with cubic time complexity $\mathcal{O}(n^3)$, e.g., MKKM-MR, SwMC, ONKC, SPMKC, and CAGL, our TFMKC with linear complexity $\mathcal{O}(n)$ exhibits obvious efficiencies, especially on larger scale datasets.
2) Comparing the late fusion MKC baselines, that is, LFMKC and OPLF, the proposed TFMKC costs more "running time," mainly caused by our partition fusion

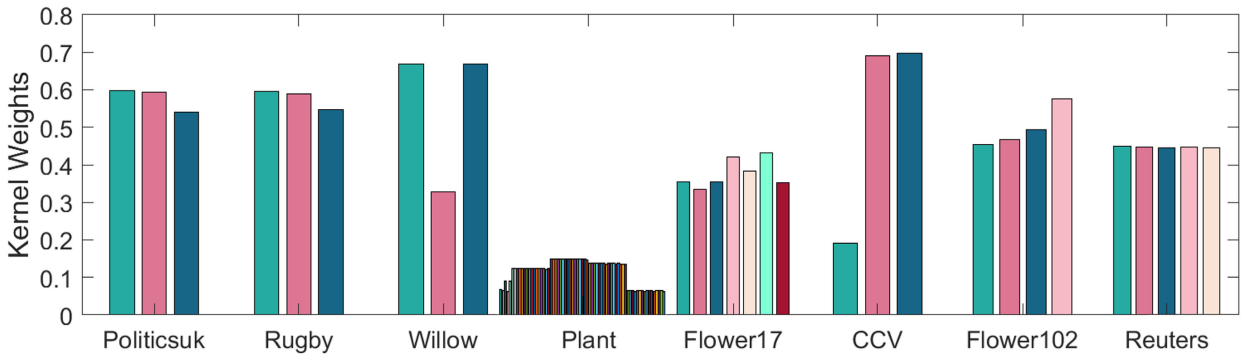


Fig. 3. Kernel weights distribution on eight datasets.

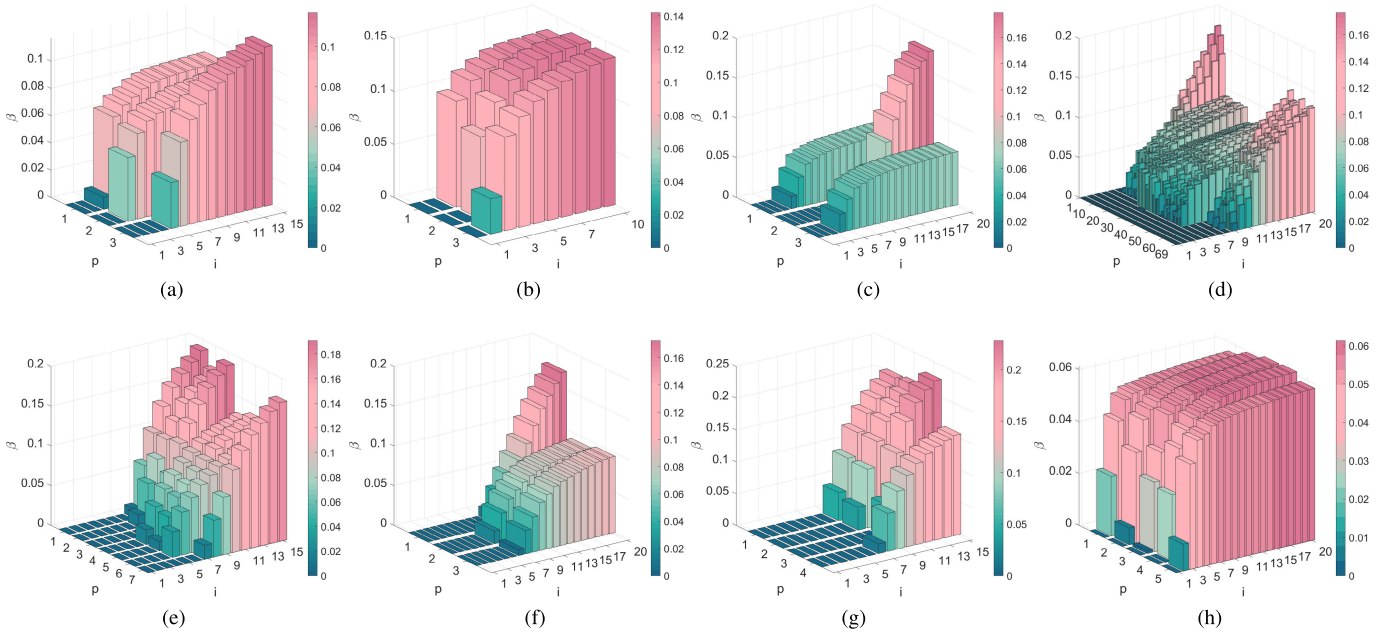


Fig. 4. Learned diverse partition weight $\boldsymbol{\beta}$ of the ith augmented kernel partition matrix in the pth view. (a) Politicsuk. (b) Rugby. (c) Willow. (d) Plant. (e) Flower17. (f) CCV. (g) Flower102. (h) Reuters.

mechanism. However, since LFMKC and OPLF involve fine-tuning or repetitive computations, their "tuning time" is much longer than ours.

3) Although our proposed model requires more CPU time than AKKM and MKKM with cubic complexity $\mathcal{O}(n^3)$, which our complex solver causes, we believe that this extra computational cost is worthwhile for significant improvement.
4) Table V reports running time comparisons on four recent baselines. SimpleMKKM is a nonparameter method without parameter tuning, yet it still requires much more run time than our model due to its complex optimization. Although FAMKKM achieves a comparable run time as our TFMKC, it requires two hyperparameters to fine-tune its performance, inducing respective parameter tuning. Both LSWMKC and MPS require more run time compared to our proposed model, and their tune time is a huge cost. By contrast, our TFMKC gets rid of the parameter tuning and achieves competitive efficiency.

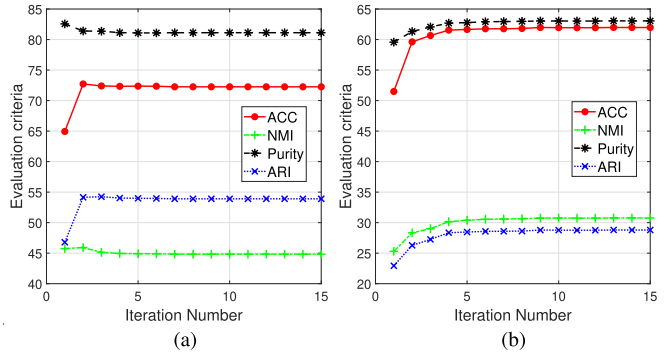


Fig. 5. Evolution of four metrics with iterations. (a) Politicsuk. (b) Plant.

G. Kernel Weights

Figs. 3 and 4 depict the learned view-wise weight $\boldsymbol{\omega}$ and partition weight $\boldsymbol{\beta}$. We observe that the nonsparse distribution varies with different views, demonstrating the superiority of our view-level fusion method in fusing diverse and complementary structures. As a result, our TFMKC adaptively achieves view-level fusion.

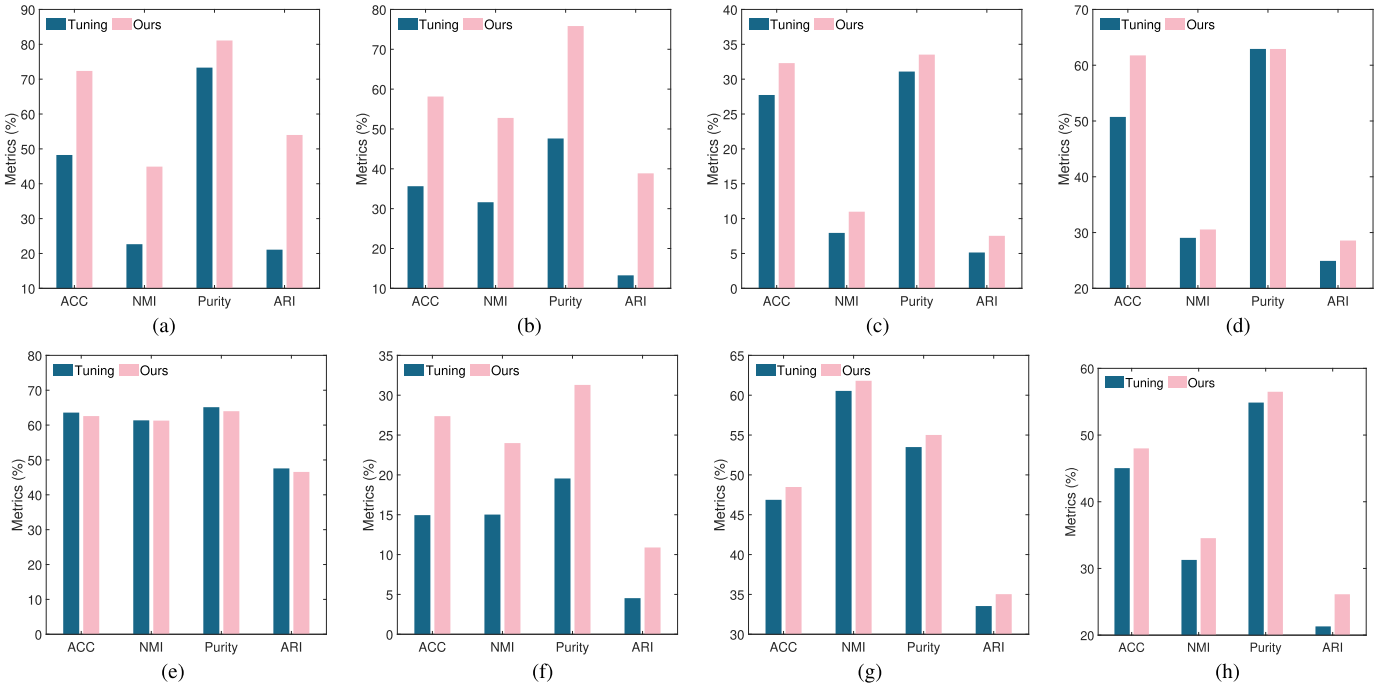


Fig. 6. Comparing clustering metrics of our tuning-free TFMKC and the fine-tuning TFMKC-tune. (a) Politicsuk. (b) Rugby. (c) Willow. (d) Plant. (e) Flower17. (f) CCV. (g) Flower102. (h) Reuters.

TABLE VI
COMPARING THE EFFICIENCY OF OUR TUNING-FREE TFMKC AND FINE-TUNING VERSION (TFMKC-TUNE)

Time (s)	Methods \ Datasets	Politicsuk	Rugby	Willow	Plant	Flower17	CCV	Flower102	Reuters
Tuning	TFMKC-tune	26.50	49.89	77.74	4790.53	367.90	4455.21	27231.36	15890.05
	TFMKC	-	-	-	-	-	-	-	-
Running	TFMKC-tune	0.59	1.66	1.30	3.47	3.50	74.25	453.86	158.90
	TFMKC	0.12	0.57	0.50	4.69	4.36	76.82	489.89	149.36

H. Evaluation of Clustering Performance

Fig. 5 records the evolution of clustering metrics on Politicsuk and Plant datasets. We observe that four metrics increase with iteration and remain stable, indicating the effectiveness of our TFMKC optimization process.

I. Ablation Study of Partition Fusion Mechanism

This section provides an ablation study of our diverse partition fusion mechanism, i.e., comparing the fine-tuning strategy (TFMKC-tune) with our tuning-free TFMKC model. TFMKC-tune involves parameter tuning, as noted in (9).

Fig. 6 and Table VI compare clustering performance and CPU time. From the table, we observe that: 1) in terms of effectiveness, although our TFMKC exhibits comparable results with TFMKC-tune on Flower17, our model achieves significant improvements on other datasets, with a large margin of 24.12%, 22.52%, 4.57%, 11.03%, 12.42%, 1.61%, and 2.96% of ACC, respectively, indicating the metric improvements of our TFMKC and 2) in terms of efficiency, our model requires much shorter CPU time due to our elegant tuning-free fusion mechanism and linear algorithm time complexity.

In addition, we plot the distribution of partition weights $\boldsymbol{\beta}$ with respect to views' ID m and search region parameter d. As our analysis of (9), the weight distribution of TFMKC-tune is a sparse trivial solution, i.e., the weight of the largest objective f_{eval} will be imposed 1, while the others are 0, which means that the potential rich benefits across diverse partitions are ignored. In contrast, our tuning-free fusion mechanism automatically reweights and fuses various partitions, which can explain significant performance improvements.

Overall, our TFMKC shows significant superiority over the traditional fine-tuning strategy.

J. Convergence

According to Section III-D, our model theoretically converges to a local optimum solution. Fig. 7 presents experimental validation on CCV and Flower102 datasets. The objective of our TFMKC decreases monotonically and converges in less than ten iterations.

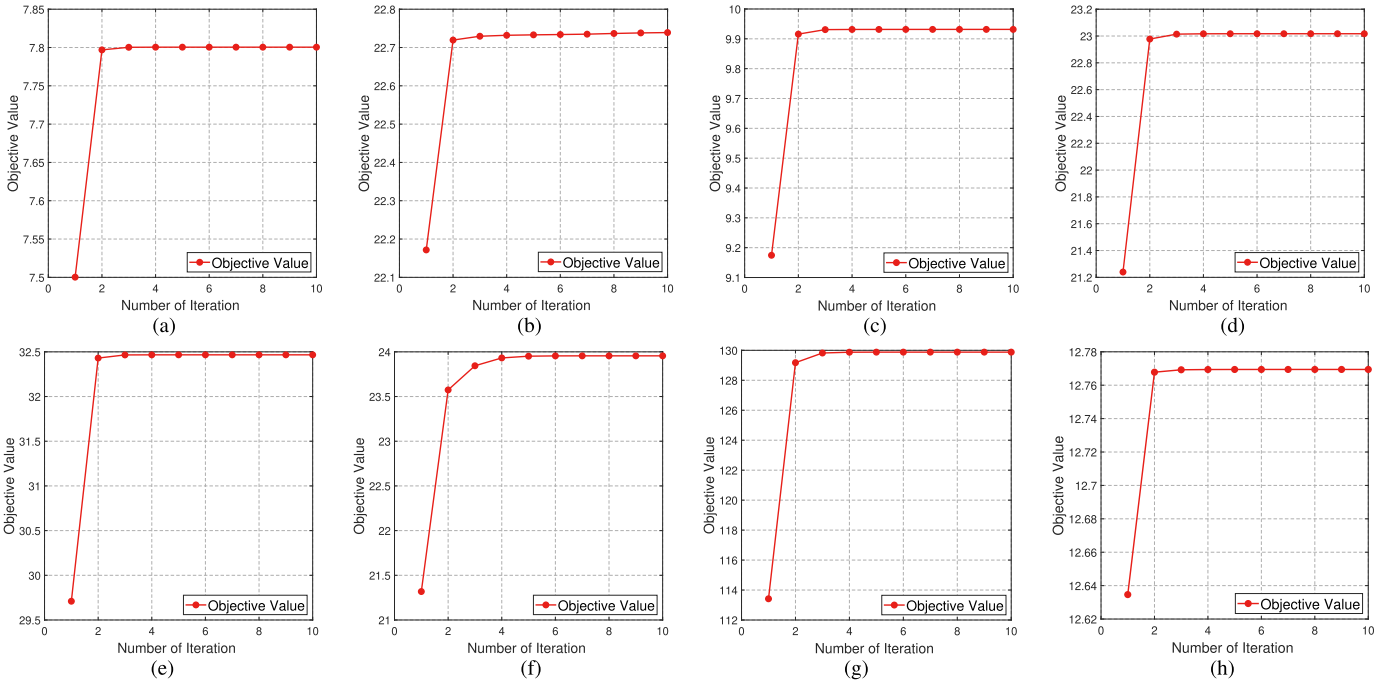


Fig. 7. Empirical validation of convergence. The objective increases monotonically and converges in less than ten iterations. (a) Politicsuk. (b) Rugby. (c) Willow. (d) Plant. (e) Flower17. (f) CCV. (g) Flower102. (h) Reuters.

V. Conclusion

This article investigates a critical issue of how to efficiently exploit diverse and complementary structures in late fusion MKC to address the limited feature representation capability of existing methods. To this end, we propose to transform the original challenging problem that directly determines the optimal partition/parameter into a flexible partition/parameter fusion problem. Based on this, we design a novel tuning-free fusion mechanism that adaptively reweights the importance of diverse partitions through optimization, integrating view- and partition-level expressive information, contributing to learning an optimal consensus partition. In addition, we provide an upper bound of our model. Extensive experiments verify our significant effectiveness and efficiency. Moreover, our flexible fusion mechanism provides a promising strategy and can be easily extended to existing MVC algorithms.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] Y. Liu et al., "Simple contrastive graph clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jun. 27, 2023, doi: 10.1109/tnnls.2023.3271871.

[3] Y. Liu, S. Zhou, X. Liu, W. Tu, and X. Yang, "Improved dual correlation reduction network," 2022, *arXiv:2202.12533*.

[4] Y. Liu et al., "Hard sample aware network for contrastive deep graph clustering," in *Proc. AAAI Conf. Artif. Intell.*, 2023, vol. 37, no. 7, pp. 8914–8922.

[5] Y. Liu et al., "Dink-net: Neural clustering on large graphs," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2023, pp. 21794–21812.

[6] J. Lu, F. Nie, R. Wang, and X. Li, "Fast multiview clustering by optimal graph mining," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Mar. 23, 2023, doi: 10.1109/tnnls.2023.3256066.

[7] C. Zhang, Y. Cui, Z. Han, J. T. Zhou, H. Fu, and Q. Hu, "Deep partial multi-view learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 5, pp. 2402–2415, May 2022.

[8] X. Peng, Y. Li, I. W. Tsang, H. Zhu, J. Lv, and J. T. Zhou, "XAI beyond classification: Interpretable neural clustering," *J. Mach. Learn. Res.*, vol. 23, p. 6, Feb. 2022.

[9] 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.

[10] L. Li et al., "BGAE: Auto-encoding multi-view bipartite graph clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 8, pp. 3682–3696, Aug. 2024.

[11] P. Zhang et al., "Let the data choose: Flexible and diverse anchor graph fusion for scalable multi-view clustering," in *Proc. 37th AAAI Conf. Artif. Intell.*, Washington, DC, USA, 2023, vol. 37, no. 9, pp. 11262–11269.

[12] X. Wan, B. Xiao, X. Liu, J. Liu, W. Liang, and E. Zhu, "Fast continual multi-view clustering with incomplete views," *IEEE Trans. Image Process.*, vol. 33, pp. 2995–3008, 2024.

[13] Z. Ren and Q. Sun, "Simultaneous global and local graph structure preserving for multiple kernel clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 5, pp. 1839–1851, May 2021.

[14] X. Liu, "SimpleMKKM: Simple multiple kernel K-means," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 5174–5186, Apr. 2023.

[15] Y. Yao, Y. Li, B. Jiang, and H. Chen, "Multiple kernel k-means clustering by selecting representative kernels," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 11, pp. 4983–4996, Nov. 2021.

[16] Y. Tang, Z. Pan, X. Hu, W. Pedrycz, and R. Chen, "Knowledge-induced multiple kernel fuzzy clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 12, pp. 14838–14855, Dec. 2023.

[17] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. Cambridge, U.K.: Cambridge Univ. Press, May 2004.

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

[19] 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.

[20] R. Wang, J. Lu, Y. Lu, F. Nie, and X. Li, "Discrete and parameter-free multiple kernel k-means," *IEEE Trans. Image Process.*, vol. 31, pp. 2796–2808, 2022.

[21] S. Zhou et al., "Multiple kernel clustering with compressed subspace alignment," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 1, pp. 252–263, Jan. 2023.

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

[23] Z. Chi et al., "Multiple kernel subspace learning for clustering and classification," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 7, pp. 7278–7290, Jul. 2023.

[24] S. Wang et al., "Multi-view clustering via late fusion alignment maximization," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Macao, China, Aug. 2019, pp. 3778–3784.

[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] 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.

[27] X. Liu et al., "One pass late fusion multi-view clustering," in *Proc. 38th Int. Conf. Mach. Learn. (ICML)*, vol. 139, Aug. 2021, pp. 6850–6859.

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

[29] J. Wang, Z. Li, C. Tang, S. Liu, X. Wan, and X. Liu, "Multiple kernel clustering with adaptive multi-scale partition selection," *IEEE Trans. Knowl. Data Eng.*, early access, May 13, 2024, doi: 10.1109/tkde.2024.3399738.

[30] J. Zhang et al., "Multiple kernel clustering with dual noise minimization," in *Proc. 30th ACM Int. Conf. Multimedia*, 2022, pp. 3440–3450.

[31] Y. Zhao and X. Li, "Spectral clustering with adaptive neighbors for deep learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 4, pp. 2068–2078, Apr. 2023.

[32] R. Wang, J. Yan, and X. Yang, "Unsupervised learning of graph matching with mixture of modes via discrepancy minimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 8, pp. 10500–10518, Mar. 2023.

[33] X. Li, Y. Zhang, and R. Zhang, "Self-weighted unsupervised LDA," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 3, pp. 1627–1632, Mar. 2023.

[34] L. Li, J. Zhang, S. Wang, X. Liu, K. Li, and K. Li, "Multi-view bipartite graph clustering with coupled noisy feature filter," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 12, pp. 12842–12854, Apr. 2023.

[35] P. Zhou, L. Du, X. Liu, Y.-D. Shen, M. Fan, and X. Li, "Self-paced clustering ensemble," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 4, pp. 1497–1511, Apr. 2021.

[36] D. Huang, C.-D. Wang, J.-S. Wu, J.-H. Lai, and C.-K. Kwoh, "Ultra-scalable spectral clustering and ensemble clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 6, pp. 1212–1226, Jun. 2020.

[37] Z. Tao, J. Li, H. Fu, Y. Kong, and Y. Fu, "From ensemble clustering to subspace clustering: Cluster structure encoding," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 5, pp. 2670–2681, May 2023.

[38] G. Tzortzis and A. Likas, "The global kernel k-means algorithm for clustering in feature space," *IEEE Trans. Neural Netw.*, vol. 20, no. 7, pp. 1181–1194, Jul. 2009.

[39] H. Q. Minh, P. Niyogi, and Y. Yao, "Mercer's theorem, feature maps, and smoothing," in *Proc. 19th Int. Conf. Comput. Learn. Theory (COLT)* (Lecture Notes in Computer Science), Pittsburgh, PA, USA, vol. 4005, G. Lugosi and H. U. Simon, Eds. Berlin, Germany: Springer, 2006, pp. 154–168, doi: 10.1007/11776420_14.

[40] C. C. Fowlkes, S. J. Belongie, F. R. K. Chung, and J. Malik, "Spectral grouping using the Nyström method," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 214–225, Feb. 2004, doi: 10.1109/TPAMI.2004.1262185.

[41] B. Wang, X. Zhang, S. Xing, C. Sun, and X. Chen, "Sparse representation theory for support vector machine kernel function selection and its application in high-speed bearing fault diagnosis," *ISA Trans.*, vol. 118, pp. 207–218, Dec. 2021.

[42] W. Yan, Y. Li, and M. Yang, "Towards deeper match for multi-view oriented multiple kernel learning," *Pattern Recognit.*, vol. 134, Feb. 2023, Art. no. 109119.

[43] F. Nie, X. Wang, C. Deng, and H. Huang, "Learning a structured optimal bipartite graph for co-clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–10.

[44] 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.

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

[46] X. Liu, Y. Dou, J. Yin, L. Wang, and E. Zhu, "Multiple kernel k-means clustering with matrix-induced regularization," in *Proc. 30th AAAI Conf. Artif. Intell.*, Phoenix, AZ, USA, 2016, pp. 1888–1894.

[47] F. Nie, J. Li, and X. Li, "Self-weighted multiview clustering with multiple graphs," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Melbourne, VIC, Australia, Aug. 2017, pp. 2564–2570.

[48] X. Liu et al., "Optimal neighborhood kernel clustering with multiple kernels," in *Proc. 31st AAAI Conf. Artif. Intell.*, San Francisco, CA, USA, 2017, pp. 2266–2272.

[49] 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.

[50] C. Zhang et al., "Generalized latent multi-view subspace clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 1, pp. 86–99, Jan. 2020.

[51] S. Huang, Z. Kang, and Z. Xu, "Auto-weighted multi-view clustering via deep matrix decomposition," *Pattern Recognit.*, vol. 97, Jan. 2020, Art. no. 107015.

[52] 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, 2023.

[53] J. Wen, Y. Xu, and H. Liu, "Incomplete multiview spectral clustering with adaptive graph learning," *IEEE Trans. Cybern.*, vol. 50, no. 4, pp. 1418–1429, Apr. 2020.

Junpu Zhang received the bachelor's degree from the Ocean University of China, Qingdao, China, in 2020, and the master's degree from the National University of Defense Technology (NUDT), Changsha, China, in 2022, where he is currently pursuing the Ph.D. degree.

His current research interests include kernel learning, graph learning, and multiview clustering.

Liang Li received the bachelor's degree from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2018, and the master's degree from the National University of Defense Technology (NUDT), Changsha, China, in 2020, where he is currently pursuing the Ph.D. degree.

He is currently a Visiting Ph.D. Student with the Centre for Frontier AI Research, A*STAR, Singapore. His research interests include graph learning and self-supervised learning.

Pei Zhang received the bachelor's degree from Yunnan University, Kunming, China, in 2018, and the master's degree from the National University of Defense Technology (NUDT), Changsha, China, in 2020, where she is currently pursuing the Ph.D. degree.

Her current research interests include incomplete multiview clustering and deep clustering.

Yue Liu (Member, IEEE) received the bachelor's degree from Northeastern University, Shenyang, China, in 2021. He is currently pursuing the master's degree with the National University of Defense Technology, Changsha, China.

His current research interests include graph neural networks, deep clustering, and self-supervised learning.

Siwei Wang (Member, IEEE) is currently an Assistant Research Professor with the Intelligent Game and Decision Laboratory (IGDL), Beijing, China. He has published several papers. His current research interests include kernel learning, unsupervised multiple-view learning, scalable clustering, and deep unsupervised learning.

Dr. Wang served as a PC Member/Reviewer in top journals and conferences, such as IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON MULTIMEDIA, ICML, CVPR, ECCV, ICCV, AAAI, and IJCAI.

Changbao Zhou received the bachelor's and master's degrees from Jilin University (JLU), Changchun, China, in 2018 and 2021, where he is pursuing the Ph.D. degree.

His research interests include hardware reliability, machine learning robustness, and security.

Xinwang Liu (Senior Member, IEEE) received the Ph.D. degree from the National University of Defense Technology (NUDT), Changsha, China, in 2013.

He is currently a Full Professor with the School of Computer, NUDT. His current research interests include kernel learning and unsupervised feature learning. He has published more than 100 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, ICML, NeurIPS, ICCV, CVPR, AAAI, and IJCAI.

Dr. Liu serves as an Associated Editor for *Information Fusion Journal*, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, and IEEE TRANSACTIONS ON CYBERNETICS. More information can be found at https://xinwangliu.github.io.

En Zhu received the Ph.D. degree from the National University of Defense Technology (NUDT), Changsha, China, in 2005.

He is currently a Professor with the School of Computer Science, NUDT. His main research interests include pattern recognition, image processing, machine vision, and machine learning. He has published more than 60 peer-reviewed articles, including IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, PR, AAAI, and IJCAI.

Dr. Zhu was awarded China National Excellence Doctoral Dissertation.