Multiple Kernel Clustering with Compressed Subspace Alignment

Sihang Zhou*, Qiyuan Ou*, Xinwang Liu† *Senior Member, IEEE,* Siqi Wang, Luyan Liu, Siwei Wang, En Zhu, Jianping Yin, and Xin Xu†

***Abstract*—Multiple kernel clustering (MKC) has recently achieved remarkable progress in fusing multi-source information to boost the clustering performance. However, the $\mathcal{O}(n^2)$ memory consumption and $\mathcal{O}(n^3)$ computational complexity prohibit these methods from being applied into median or large-scale applications, where n denotes the number of samples. To address these issues, we carefully redesign the formulation of subspace segmentation-based MKC, which reduces the memory and computational complexity to $\mathcal{O}(n)$ and $\mathcal{O}(n^2)$, respectively. The proposed algorithm adopts a novel sampling strategy to enhance the performance and accelerate the speed of MKC. Specifically, we first mathematically model the sampling process and then learn it simultaneously during the procedure of information fusion. By this way, the generated anchor point set can better serve data reconstruction across different views, leading to improved discriminative capability of the reconstruction matrix and boosted clustering performance. Although the integrated sampling process makes the proposed algorithm less efficient than the linear complexity algorithms, the elaborate formulation makes our algorithm straightforward for parallelization. Through the acceleration of GPU and multi-core techniques, our algorithm achieves superior performance against the compared state-of-the-art methods on six datasets with comparable time cost to the linear complexity algorithms.**

***Index Terms*—Multiple Kernel Clustering, Compressed Subspace Alignment, Sampling Process Modeling.**

I. Introduction

MULTIPLE kernel clustering (MKC) [1], which dexterously integrates heterogeneous information from multiple base kernels to improve clustering performance, has attracted intensive attention of many researchers and witnessed a soaring improvement in the past few years. According to the information fusion mechanism, the existing literature of MKC in this field can be roughly divided into three categories, i.e., linear combination-based methods [1]–[5], consensus information extraction-based methods [6]–[14], and co-training-based methods [15], [16]. Among these methods, the first category of algorithms model the information fusion process as a linear combination problem. They assume that the optimal kernel lies in the linear space extended by the base kernels and integrate multi-source information by finding the optimal linear combination weights of base kernels. The second category of methods decompose the base kernels into the sum of a shared cluster structure indicating matrix and distinct perturbation matrices. By doing this, the underlying consistent geometric information is extracted and enhanced. According to the third category of methods, each base kernel is with sufficient information and can conduct predictions independently. After the predictions are acquired, the co-training algorithms amplify the prediction values which are consistent across views while modify the inconsistent ones by referring to the predictions which are with higher confidence.

Although remarkable improvement has been made, the large memory consumption of base kernel matrices (i.e., $\mathcal{O}(n^2)$) and the high computational complexity (i.e., $\mathcal{O}(n^3)$) of the corresponding optimization algorithms limit the utility of these algorithms in practical applications [17]–[21]. To address these issues, a large number of methods have been proposed in the existing literature, which can be roughly divided into four categories. The first category of methods project features from different views into a collaborative Hamming space and concurrently learns binary codes together with the cluster structures within one light-weighted framework [22]–[26]. In these methods, short binary codes and fast bit-operations are adopted for data storage and cluster structures optimization, making them computational and storage friendly. Instead of processing all data simultaneously, the second category of methods [27]–[29] solve the large scale cluster optimization problem in an online learning fashion. Since only a part of the data [27] (or views [28]) are processed at a learning step, these algorithms are usually with low computational and storage complexities. The third category of methods find that the main computational consumption of the existing algorithms lies in information fusion and the consensus partition matrix computation procedure. In order to reduce the consumption, deep learning-based algorithms directly train a deep neural network to regress the MKC cluster indicating matrix on a small subset, and then estimate the matrix of the whole data set using the trained network. Finally, a k-means algorithm is conducted on the estimated matrix for clustering. In this way, the cluster indicating matrix is also learned efficiently. Recently, the fourth category of methods that adopt a sampling strategy for efficient clustering have attracted the attention of a multitude of researchers. In these methods, the authors

S. Zhou is with the School of Computer, National University of Defense Technology, Changsha 410073, P.R. China. He is also with the College of Intelligence Science and Technology, National University of Defense Technology, Changsha 410073, China. (E-mail: sihangjoe@gmail.com)

Q. Ou, X. Liu, S. Wang, E. Zhu are with the School of Computer, National University of Defense Technology, Changsha 410073, P.R. China. (E-mail: xinwangliu@nudt.edu.cn)

L. Liu is with the Jarvis Lab, Tencent, Shenzhen 518057, China

J. Yin is with the School of Cyberspace Science, Dongguan University of Technology, Guangdong 523808, China.

X. Xin is with the College of Intelligence Science and Technology, National University of Defense Technology, Changsha 410073, China. (xinxu@nudt.edu.cn)

* Equal contribution. † Corresponding authors.

assume that the manifold of a dataset can be sufficiently represented by sampling only a small subset. Specifically, in [30]–[33], a novel bipartite and a sparse affinity matrix that only record the similarity between the selected salient point set and the data points are learned for clustering, respectively. In efficient multi-view subspace clustering [34], a compact reconstruction matrix which reconstructs the data points with only the pre-learned anchor points can be generated with linear time complexity. In these methods, the sampling technique has largely improved the learning speed without a significant loss of the clustering accuracy.

Although various improvements have been achieved by the literature, we observe that the algorithms from the fourth category suffer from the following drawbacks. First, the anchor points in these methods are generated by a k-means clustering or a random sampling operation, which is isolated from the process of multi-view information fusion. Although this setting makes the clustering process extremely fast (with linear complexity), this also makes the learned anchor points less suitable for downstream tasks such as spectral clustering or subspace clustering. Secondly, in these methods, an independent anchor point set is generated without information exchange among different views, which could under-fit the overall structure of the multi-view data. Both factors could adversely affect the discriminative capability of the learned affinity matrix or the reconstruction matrix, leading to unsatisfactory clustering performance.

In this paper, we propose a compressed subspace alignment-based multiple kernel clustering (CSA-MKC) algorithm to solve the above problems. Specifically, in our method, three configurations are proposed to find an appropriate balance between clustering speed and accuracy. **First**, we mathematically formulate the sampling process and subtly integrate it into the process of multi-view subspace clustering. This setting allows the two processes to negotiate, in order to best serve each other in a united system. Specially, in our formulation, a consensus sampling matrix that fuses the information from all base kernels is learned to make the generated affinity matrix to be more discriminative for MKC. **Second**, a late fusion technique [35], [36] is adopted to reduce both storage and computational cost of the proposed algorithm. **Third**, we reformulate the target of the subspace clustering as subspace alignment to further reduce the complexity of optimization. The contributions of this paper are summarized as follows:

i) Our MKC method, for the first time, integrates sampling into multi-view clustering and learns the two processes iteratively in a unified framework, which makes the learned anchor point set better serve the need of clustering.

ii) We propose a late-fusion based multiple kernel clustering algorithm with $\mathcal{O}(n)$ storage consumption and $\mathcal{O}(n^2)$ computational complexity. Although the proposed algorithm is computationally more costly than the linear complexity algorithms, it achieves a better balance between speed and clustering accuracy.

iii) Since the computational bottleneck of the proposed algorithm lies in matrix multiplication, which is suitable for parallelization, through the acceleration of GPU, our proposed algorithm outperforms the compared state-of-the-art methods with comparable efficiency against the linear complexity algorithms.

II. Related Work

A. Notion

For the clarity of the paper, we first clarify the definition of some of the variables in Table I. We denote scalars, vectors and matrices using lower-case, bold lower-case and bold upper-case letters, e.g., n, $\mathbf{x}$, and $\mathbf{X}$.

TABLE I: Summary of notations

k	The number of data clusters.
p	The number of base kernels.
l	The number of anchor points.
$\boldsymbol{\mu} \in \mathbb{R}^p$	The vector of kernel combination weights.
$\mathbf{K}_i \in \mathbb{R}^{n\times n}$	The i-th base kernel.
$\mathbf{K}_{\boldsymbol{\mu}} \in \mathbb{R}^{n\times n}$	The combined kernel according to $\boldsymbol{\mu}$.
$\mathbf{I}_n \in \mathbb{R}^{n\times n}$	The n-th order identity matrix.
$\mathbf{H}^* \in \mathbb{R}^{k\times n}$	The partition matrix of kernel clustering.
$\mathbf{P} \in \mathbb{R}^{n\times l}$	The sampling matrix.
$\mathbf{S}_i \in \mathbb{R}^{n\times l}$	The reconstruction matrix of the i-th partition matrix.
$\mathbf{S} \in \mathbb{R}^{n\times l}$	The consensus reconstruction matrix.

B. Multiple Kernel K-means

Given a collection of p base kernels $\{\mathbf{K}_i\}_{i=1}^p \in \mathbb{R}^{n\times n}$ which are generated through calculating the similarity over n samples $\{\mathbf{x}_i\}_{i=1}^n \in \mathbb{R}^d$ with p mapping functions $\{\phi_i(\cdot)\}_{i=1}^p$: $\mathbf{x} \in \mathcal{X} \rightarrow \mathcal{H}$. Here, $\phi_i(\cdot)$ is a mapping function that maps $\mathbf{x}$ onto a reproducing kernel Hilbert space $\mathcal{H}_i$ and d is the dimension of sample $\mathbf{x}$. The objective of multiple kernel k-means clustering is to minimize the clustering distortion in the kernel space extended by the base kernel functions. In this framework, the optimal base kernel combination coefficients and the sample partition matrix are learned simultaneously. Through simple deduction, the formulation of this problem can be concisely written as:

$$\begin{aligned} &\min_{\mathbf{H}\in\mathbb{R}^{k\times n},\boldsymbol{\mu}\in\mathbb{R}_+^p} \operatorname{Tr}(\mathbf{K}_{\boldsymbol{\mu}}(\mathbf{I}_n - \mathbf{H}^{*\top}\mathbf{H}^*)) \\ &s.t.\ \mathbf{H}^*\mathbf{H}^{*\top} = \mathbf{I}_k,\ \boldsymbol{\mu}^\top \mathbf{1}_p = 1. \end{aligned} \tag{1}$$

Here, $\boldsymbol{\mu} = [\mu_1, \mu_2, \cdots, \mu_p]$ denotes the kernel combination weights of each base kernel and $\mathbf{K}_{\boldsymbol{\mu}} = \sum_{i=1}^p \mu_i^2 \mathbf{K}_i$ is the combined kernel which has integrated information from different views. $\mathbf{H}^* \in \mathbb{R}^{k\times n}$ is the cluster partition matrix, where k is the cluster number. The target of the Eq.(1) can be easily optimized by conducting an iterative optimization procedure, in which a quadratic programming problem and a singular value decomposition (SVD) are done in turn to have $\boldsymbol{\mu}$ and $\mathbf{H}^*$ optimized gradually. Although the optimization process is clear, simple and is guaranteed to converge to a local optimal solution, the $\mathcal{O}(n^2)$ storage consumption of base kernels and the $\mathcal{O}(n^3)$ computational complexity of SVD limit the algorithm from scaling to large datasets [9]. To accelerate the clustering speed and reduce the memory cost, many researchers in the field turn to a late-fusion for multiple kernel clustering.

C. Late Fusion-based Multiple Kernel Clustering

Instead of using base kernels to represent sample distribution in each view ($\{\mathbf{K}_i\}_{i=1}^p \in \mathbb{R}^{n\times n}$), late fusion-based multiple kernel clustering [35], [37] uses a more compact fashion (data partition matrices $\{\mathbf{H^*}_i\}_{i=1}^p \in \mathbb{R}^{k\times n}$) for structure representation. Moreover, since it is discovered in a recent study that the target of k-means clustering is conceptually equivalent to maximizing the alignment between base partitions and the consensus partition [36], researchers are able to design the following late fusion-based MKC formulation:

$$\begin{aligned}&\max_{\mathbf{H}^*,\{\mathbf{W}_i\}_{i=1}^p,\boldsymbol{\beta}} \mathrm{Tr}(\mathbf{H}^{*\top}\mathbf{H}_{\boldsymbol{\beta}} + \lambda\mathbf{H}^{*\top}\mathbf{N}),\\ &s.t.\ \mathbf{H}^*\mathbf{H}^{*\top} = \mathbf{I}_k,\ \mathbf{W}_i^\top\mathbf{W}_i = \mathbf{I}_k, \|\boldsymbol{\beta}\|_2 = 1, \beta_i \geq 0,\end{aligned} \quad (2)$$

where $\{\mathbf{W}_i\}_{i=1}^p \in \mathbb{R}^{k\times k}$ are a set of rotation matrices, $\mathbf{H}_{\boldsymbol{\beta}} = \sum_{i=1}^p \beta_i\mathbf{W}_i\mathbf{H^*}_i$ is a linear combination of the rotated partitions generated from each base kernel, $\mathbf{N} \in \mathbb{R}^{k\times n}$ denotes the average cluster indicating matrix and λ is a trade-off parameter. In this formulation, the authors intend to maximize the alignment between the optimal data partition $\mathbf{H}^* \in \mathbb{R}^{k\times n}$ and both the linear combination of the rotated base partitions as well as the average partition. Through careful deduction and optimization the authors are able to achieve satisfactory clustering performance within linear computational complexity, indicating promising potential of the late fusion-based learning mechanism on speeding up the MKC algorithms while maintaining their performance.

D. Subspace Clustering

Subspace clustering [38] is a series of methods which cluster high-dimensional data points by revealing the low-dimensional subspaces extended by samples from different clusters. The underlying assumption of these methods is that samples from the same cluster can be self-reconstructed by each other. Given a set of data vectors $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n]$, by further considering the noise, outliers or missing information within data, a common formulation of many popular methods in this branch is:

$$\min_{\mathbf{S},\mathbf{E}} \|\mathbf{S}\|_\dagger + \lambda\|\mathbf{E}\|_\ddagger, \quad \text{s.t. } \mathbf{X} = \mathbf{X}\mathbf{S} + \mathbf{E}, \quad (3)$$

where $\mathbf{S} \in \mathbb{R}^{n\times n}$ and $\mathbf{E} \in \mathbb{R}^{n\times n}$ are the reconstruction matrix and error indicating matrix, respectively. In the equality constraint, sample matrix $\mathbf{X}$ is required to reconstruct it self with an error matrix $\mathbf{E}$. In the target formulation, different $\|\cdot\|_\dagger$ and $\|\cdot\|_\ddagger$ are introduced to add various prior knowledge to $\mathbf{Z}$ and $\mathbf{E}$. Different kinds of norms like, ℓ_1-norm, nuclear norm, Frobenius norm are corresponding to different properties like sparse, low-rank and block diagonal of the respective matrices, respectively. [39], [40]. Since subspace clustering is able to extract intrinsic cluster structure among data against noise and outliers, it is also suitable for the multiple kernel clustering scenario.

E. MKC with Subspace Clustering

To extend subspace clustering algorithms to multi-view clustering circumstances, in [12], Zhou et. al propose to learn the optimal subspace reconstruction matrix w.r.t. the optimal linear combined kernel $\mathbf{K}_{\boldsymbol{\beta}}$:

$$\begin{aligned}&\min_{\mathbf{Z},\boldsymbol{\beta}} \|\mathbf{K}_\beta - \mathbf{K}_\beta\mathbf{Z}\|_F^2 + \alpha\|\mathbf{Z}\|_F^2 + \gamma\boldsymbol{\beta}^\top\mathbf{M}\boldsymbol{\beta},\\ &s.t. rank(\mathbf{Z}) = r; \boldsymbol{\beta} \geq 0, \|\boldsymbol{\beta}\|_1 = 1; \mathbf{K}_\beta = \sum\nolimits_{i=1}^p \beta_i\mathbf{K}_i.\end{aligned} \quad (4)$$

In the target function, r is the target rank number, the first term is the kernel reconstruction term. It requires to minimize the self-reconstruction error of the linear combined optimal kernel. The second term is the regularization term that encourages $\mathbf{Z}$ to better preserve block diagonal structure among data. The third term is the diversity induction term. In this term, matrix $\mathbf{M}$ is:

$$M_{a,b} = \frac{\langle\mathbf{K}_a, \mathbf{K}_b\rangle_{\mathrm{F}}}{\|\mathbf{K}_a\|_{\mathrm{F}}\|\mathbf{K}_b\|_{\mathrm{F}}}, \quad (5)$$

where $\langle\cdot,\cdot\rangle_{\mathrm{F}}$ is the trace operation and $\mathbf{M}$ denotes the centered kernel alignment-based correlation [41] of the base neighbor-kernels. In the constraints, the first term is the exact rank constraint, the second term is the combination coefficient constraint. Although good performance has been achieved by the existing algorithms in many applications, the large storage and computational consumption cost by the complex operations on the $n \times n$ reconstruction matrix $\mathbf{Z}$ limits the efficiency of these methods.

F. Sampling-based Efficient Clustering Algorithms

Sampling has long been a hotspot technique which is widely used in efficient multi-view spectral clustering [30] and multi-view subspace clustering [34], etc. In these methods, to avoid doing complex operations on computational and storage inefficient $n\times n$ affinity graphs, researchers select only a small number of the instances as anchors then efficiently learn a sub-graph $\mathbf{S} \in \mathbb{R}^{n\times l}$ between the anchor points and the data, where l is the number of anchor points. As was proved by these methods, the sampling operation can help largely reduce both storage and computational time while providing comparable clustering performance. However, in the existing literature, the sampling procedure is conducted isolated from the multi-view clustering process. Also, it is performed independently in each view, leading to less discriminative anchor points. To solve this problem, in the next section, we propose a compressed subspace alignment-based MKC algorithm.

III. Multiple Kernel Clustering with Compressed Subspace Alignment

In this section, we first modify the target of multi-view subspace clustering [12], [13], [42] to compressed multi-view subspace clustering, then we further simplify the formulation by introducing subspace alignment to improve the optimization efficiency. After that, we propose an efficient three-step iterative optimization algorithm with proved convergence to solve the resultant optimization problem.

A. The Proposed Formulation

1) Sampling-base subspace clustering for MKC.: In order to improve the computational and memory efficiency, our proposed algorithm modify the classic multi-view subspace clustering algorithm into a late fusion fashion which takes the cluster indicating matrices ($\{\mathbf{H}_i\}_{i=1}^p \in \mathbb{R}^{m\times n}$) instead of the original kernel matrices ($\{\mathbf{K}_i\}_{i=1}^p \in \mathbb{R}^{n\times n}$) of each view as input. Specifically, for $\forall i \in [1,p]$, $\mathbf{H}_i$ is calculated by doing SVD on the $\mathbf{K}_i$ and then take the top m eigenvectors. Specially, in our paper, we set $m = 2k$ in all our experiments. Moreover, we further model the sampling process mathematically and integrate it with multi-view information fusion. As a consequence, a much smaller sample set with better discriminative capacity is generated for sample reconstruction, leading to higher computational efficiency and potentially better clustering performance. The formulation of our algorithm is presented as follow:

$$\begin{aligned}&\min_{\mathbf{P},\mathbf{S},\{\mathbf{S}_i\}_{i=1}^p} \sum\nolimits_{i=1}^p \|\mathbf{H}_i - \mathbf{H}_i\mathbf{P}{\mathbf{S}_i}^\top\|_{\mathrm{F}}^2 + \alpha\sum\nolimits_{i=1}^p \|\mathbf{S}-\mathbf{S}_i\|_{\mathrm{F}}^2,\\ &s.t.\ 0\le\mathbf{S}_i\le 1,\ 0\le\mathbf{S}\le 1,\ \mathbf{P}^\top\mathbf{P}=\mathbf{I}_l.\end{aligned} \tag{6}$$

Here, $\mathbf{P}\in\mathbb{R}^{n\times l}$ is a sampling matrix that generates anchor points by learning l linear combinations of the data points. It is a common sampling matrix if each of its column is required to have only one element to be 1 while other elements to be 0. In our setting, to improve the representative capability of the generated anchor points and to improve their information diversity, the sampling matrix is relaxed to be an orthogonal matrix. Moreover, to make the generated anchor point set to be suitable for reconstruction across views, a consensus matrix $\mathbf{P}$ is learned for all views. $\{\mathbf{S}_i\}_{i=1}^p \in \mathbb{R}^{n\times l}$ is the reconstruction matrix of the i-th base kernel that reconstructs the data with the generated anchor points. $\mathbf{S}$ is the consensus matrix that fuses information from each view. In the target function, the reconstruction term $\mathbf{H}_i\mathbf{P}{\mathbf{S}_i}^\top$ first maps the partition matrix into a low dimensional space with the projection matrix $\mathbf{P}$. Then it recovers $\mathbf{H}_i$ with the reconstruction matrix $\mathbf{S}_i$. This is similar to the process of compressed sensing (CS). The difference is that in CS the features are the objects which are compressed, while in our setting the sample points are compressed.

2) MKC with compressed subspace alignment.: Although in Eq. (6), the introduction of late-fusion learning mechanism and the sampling matrix has largely reduced the storage and computational consumption, the re-weighted method [43] for the optimization of $\mathbf{P}$ in this setting is still complex and slow. Since in multiple kernel learning algorithms, kernel polarization that maximizes the alignment between the target kernel with the linearly combined kernel has long been taken as an alternative formulation of minimizing the difference between the two kernels [44], in our paper, to further improve the computational speed, we convert subspace clustering into subspace alignment and propose the following formulation:

$$\begin{aligned}&\min_{\mathbf{P},\mathbf{S},\{\mathbf{S}_i\}_{i=1}^p} -\sum_{i=1}^p \mathrm{Tr}(\mathbf{H}_i(\mathbf{H}_i\mathbf{P}{\mathbf{S}_i}^\top)^\top) + \alpha\sum_{i=1}^p \|\mathbf{S}-\mathbf{S}_i\|_{\mathrm{F}}^2,\\ &s.t.\ 0\le\mathbf{S}_i\le 1,\ 0\le\mathbf{S}\le 1,\ \mathbf{P}^\top\mathbf{P}=\mathbf{I}_l.\end{aligned} \tag{7}$$

In Eq. (7), we follow the setting of many existing works [36], [44] and use a simpler alignment maximization term to replace the Frobenius norm of the difference, thus further reducing the optimization difficulty.

B. Optimization Algorithm

To optimize the resultant problem in Eq. (7), we propose a three-step iterative optimization algorithm. In each step, two of the variables are fixed and the remaining one is optimized. The detailed procedure is presented as follow.

1) Update $\mathbf{P}$*:* Given $\mathbf{S}$ and $\{\mathbf{S}_i\}_{i=1}^p$, the optimization problem of Eq. (7) w.r.t. $\mathbf{P}$ becomes:

$$\max_{\mathbf{P}} \sum\nolimits_{i=1}^p \mathrm{Tr}(\mathbf{H}_i(\mathbf{H}_i\mathbf{P}{\mathbf{S}_i}^\top)^\top) \quad s.t.\ \mathbf{P}^\top\mathbf{P}=\mathbf{I}_l. \tag{8}$$

Letting $\mathbf{A} = \sum_{i=1}^p \mathbf{H}_i^\top\mathbf{H}_i\mathbf{S}_i$, Eq. (8) can be simplified as:

$$\max_{\mathbf{P}} \mathrm{Tr}(\mathbf{A}\mathbf{P}^\top) \quad s.t.\ \mathbf{P}^\top\mathbf{P}=\mathbf{I}_l. \tag{9}$$

Denote the SVD of matrix $\mathbf{A}$ as $\mathbf{U}\mathbf{D}\mathbf{V}^\top$ and the optimal solution of Eq. (9) as $\mathbf{P}^*$. It is easy to know that $\mathbf{D}$ is a diagonal nonnegative matrix and both $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices. Through deduction we can find that Eq. (9) has a closed form solution, i.e. $\mathbf{P}^* = \mathbf{U}\mathbf{V}^\top$.

Theorem 1. *The optimal solution of Eq.* (9) *is* $\mathbf{P}^* = \mathbf{U}\mathbf{V}^\top$.

Proof. First, since ${\mathbf{P}^*}^\top\mathbf{P}^* = \mathbf{I}_l$, we can easily find that $\mathbf{P}^*$ is a solution of Eq. (9). Then we prove $\mathbf{P}^*$ is the optimal solution. Substitute $\mathbf{A}$ with its SVD and $\mathbf{P}$ with $\mathbf{P}^*$, the resulting target value of Eq. (9) in this circumstance becomes:

$$\mathrm{Tr}(\mathbf{A}{\mathbf{P}^*}^\top) = \mathrm{Tr}(\mathbf{U}\mathbf{D}\mathbf{V}^\top(\mathbf{U}\mathbf{V}^\top)^\top).$$

Denote the singular values of $\mathbf{A}$ as $\delta_1, \delta_2, \cdots, \delta_n$, we have

$$\mathrm{Tr}(\mathbf{A}{\mathbf{P}^*}^\top) = \sum\nolimits_{i=1}^m \delta_i.$$

Moreover, for $\forall\mathbf{P}$, where $\mathbf{P}^\top\mathbf{P} = \mathbf{I}_m$,

$$\mathrm{Tr}(\mathbf{A}\mathbf{P}^\top) = \mathrm{Tr}(\mathbf{U}\mathbf{D}\mathbf{V}^\top\mathbf{P}^\top) = \mathrm{Tr}(\mathbf{V}^\top\mathbf{P}^\top\mathbf{U}\mathbf{D}).$$

Let $\mathbf{G} = \mathbf{V}^\top\mathbf{P}^\top\mathbf{U}$, then $\mathbf{G}\mathbf{G}^\top = \mathbf{I}_m$. As a consequence,

$$\mathrm{Tr}(\mathbf{A}\mathbf{P}^\top) = \mathrm{Tr}(\mathbf{G}\mathbf{D}) \le \sum\nolimits_{i=1}^m \delta_i.$$

Therefore, for $\forall\mathbf{P}$, if $\mathbf{P}$ is a solution of Eq. (9), we have $\mathrm{Tr}(\mathbf{A}\mathbf{P}^\top) \le \mathrm{Tr}(\mathbf{A}{\mathbf{P}^*}^\top)$. Overall, $\mathbf{P}^* = \mathbf{U}\mathbf{V}^\top$ is the optimal solution of Eq. (9). □

2) Update $\mathbf{S}_i$*:* Given $\mathbf{P}$, $\mathbf{S}$, and the $\{\mathbf{S}_j\}_{j\neq i}^p$, the optimization problem in Eq. (7) w.r.t. $\mathbf{S}_i$ reduces to the following problem:

$$\min_{\mathbf{S}_i} -\mathrm{Tr}(\mathbf{P}^\top\mathbf{H}_i^\top\mathbf{H}_i\mathbf{S}_i) + \alpha\|\mathbf{S}-\mathbf{S}_i\|_{\mathrm{F}}^2,\ s.t.\ 0\le\mathbf{S}_i\le 1. \tag{10}$$

Denote $\mathbf{B} = (\mathbf{P}^\top\mathbf{H}_i^\top\mathbf{H}_i + 2\alpha\mathbf{S}^\top)/(2\alpha)$, through simple deduction, Eq. (10) can be converted to a more compact form:

$$\min_{\mathbf{S}_i} \|\mathbf{S}_i - \mathbf{B}\|_{\mathrm{F}}^2,\ s.t.\ 0\le\mathbf{S}_i\le 1. \tag{11}$$

Denote the optimal solution of Eq. (11) as $\mathbf{S}_i^*$, it has a closed form as follow:

$$\mathbf{S}_i^* = \mathbf{Proj}_{[0,1]}(\mathbf{B}), \tag{12}$$

where, $\mathbf{Proj}_{[0,1]}(\cdot)$ is a function that projects a real matrix into the range of $[0, 1]$.

3) Update $\mathbf{S}$*:* Given $\{\mathbf{S}_i\}_{i=1}^p$ and $\mathbf{P}$, the optimization problem w.r.t. $\mathbf{S}$ becomes:

$$\min_{\mathbf{S}} \sum\nolimits_{i=1}^{p} \|\mathbf{S} - \mathbf{S}_i\|_{\mathrm{F}}^2, \ s.t.\ 0 \leq \mathbf{S} \leq 1. \tag{13}$$

Let $\mathbf{C} = \sum_{i=1}^p \mathbf{S}_i/p$, Eq. (13) is equivalent to

$$\min_{\mathbf{S}} \|\mathbf{S} - \mathbf{C}\|_{\mathrm{F}}^2, \ s.t.\ 0 \leq \mathbf{S} \leq 1. \tag{14}$$

The optimal solution of Eq. (14) is $\mathbf{S}^* = \mathbf{Proj}_{[0,1]}(\mathbf{C})$.

In sum, our algorithm for solving Eq. (7) is outlined in Algorithm 1, where $obj^{(t)}$ denotes the objective value at the t-th iteration.

Algorithm 1 Multiple Kernel Clustering with Compressed Subspace Alignment

Input:

Base cluster indicating matrices $\{\mathbf{H}_{(1)}, \ldots, \mathbf{H}_{(p)}\}$, number of clusters k, parameter α, number of anchor points l.

Output:

The consensus reconstruction matrix $\mathbf{S}$.

1: **Initialization** Generate $\mathbf{P}^{(0)}$ by normalizing and orthogonalizing a random $n \times l$ matrix. Set $\mathbf{S}^{(0)} = \mathbf{0}$ and $t = 1$.
2: **repeat**
3: Calculate $\mathbf{S}_i^{(t)}$ by optimizing Eq. (11);
4: Calculate $\mathbf{P}^{(t)}$ by optimizing Eq. (9);
5: Calculate $\mathbf{S}^{(t)}$ by optimizing Eq. (14);
6: $t = t + 1$.
7: **until** $\|\mathbf{P}^{(t)} - \mathbf{P}^{(t-1)}\|_{\mathrm{F}} / \|\mathbf{P}^{(t)}\|_{\mathrm{F}} \leq 10^{-3}$.

C. Algorithmic Discussion

1) Convergence Analysis: In our proposed three-step optimization algorithm, each sub-step has a closed-form optimal solution. As a consequence, the objective of Algorithm 1 is guaranteed to be monotonically decreased when optimizing one variable with the others fixed at each iteration. At the same time, the objective is lower-bounded since $\mathbf{P}$ is an orthogonal matrix and $0 \leq \mathbf{S}, \mathbf{S}_i \leq 1$. Since the target value in the optimization procedure is monotonously decreased with a lower bound, the convergence of our proposed algorithm is guarenteed.

2) Computational Complexity Analysis: Our optimization algorithm is composed of three sub-problems, each has a closed form solution. The overall procedure is reported in Algorithm 1. Specifically, the optimization of $\mathbf{P}$ requires doing SVD on a $n \times l$ matrix. This can be efficiently conducted with time complexity of $\mathcal{O}(nl^2)$. Since in large datasets $l \ll n$, i.e. the anchor points number is much smaller than the size of dataset, the time complexity of the SVD in this circumstance is $\mathcal{O}(n)$. Also, since the optimization of $\{\mathbf{S}_i\}_{i=1}^p$ and $\mathbf{S}$ have closed form solution, they can be optimized efficiently. The largest time consumption of the proposed algorithm comes from the matrix multiplications like calculating $\sum_{i=1}^p \mathbf{H}_i^\top \mathbf{H}_i \mathbf{S}_i$ and $\mathbf{P}^\top \mathbf{H}_i^\top \mathbf{H}_i$, they make the time complexity of the proposed algorithm becomes $\mathcal{O}(n^2)$. This is a larger time consumption compared to some of the state-of-the-art linear complexity algorithms (like MVC-LFA [36] and LMVSC [34]). As the sampling process is optimized simultaneously with the information fusion process, the extra computational consumption mainly comes from the unified optimization configuration. Although more computation is cost, the representative capacity of the learned anchor point set is also improved. With the current setting, our algorithm tries to find a good balance between computational efficiency and clustering performance. Moreover, since the operation of matrix multiplication is easy for parallelization, through careful parallel implementation, the side effect of sampling process integration can be appropriately relieved. Experimental results in the next section show that the GPU version of our algorithm can achieve comparable efficiency with the linear complexity methods.

IV. Experiment

In this section, we first carefully design and compare four subspace clustering-base multi-view clustering algorithms to construct an ablation study to verify the effectiveness of the sampling integration mechanism. In the first experiment, by illustrating the distribution of the learned anchor points of different compared algorithms, we also give light on the mechanism of the effectiveness. Then, we extensively compare the clustering performance and the computational consumption with the state-of-the-art algorithms to validate the effectiveness of the proposed algorithm. In the efficiency comparison part, both the performance of the CPU version and the GPU enhanced version were compared. Finally, we further analyze the properties like, the sensitivity and convergence of the proposed algorithm.

A. Datasets Overview and Experimental Settings

Datasets introduction. We evaluate the clustering performance of the proposed algorithm on 6 popular datasets. The detailed information of these datasets is listed in Table II. From this table, we observe that the number of samples, views and the number of categories of these datasets range in a large scale from 940 to 60,000, 3 to 69, and 3 to 102, respectively. For these datasets, the kernel matrices of the first five datasets are pre-computed with carefully designed similarity function and are publicly available from websites[1][2][3]. For the MNIST dataset[4], it is a classic handwritten digits dataset with 60,000 samples. To construct multi-view description for the samples, we adopt 3 classic ImageNet pre-trained deep neural networks, i.e. VGG [45], DenseNet121 [46] and ResNet101 [47] to extract features. With the extracted features, we finally construct 3 linear kernels for the dataset.

[1]http://mkl.ucsd.edu/dataset/protein-fold-prediction
[2]http://www.robots.ox.ac.uk/ vgg/data/flowers/
[3]http://www.vision.caltech.edu/archive.html
[4]http://yann.lecun.com/exdb/mnist/

TABLE II: Information of benchmark datasets

Datasets	# Samples	# Views	# Clusters
Plant	940	69	4
Flower17	1360	7	17
Caltech101	1530	25	102
Mfeat	2000	12	10
Flower102	8189	4	102
MNIST	60000	3	10

Experimental setting. In our experiments, the implementation of all the compared algorithms is downloaded from the authors websites. The hyper-parameters are set according to the suggestions of the corresponding literature. As to our proposed method, the regularization parameter and the number of anchor points are chosen in the range of $[2^{-15}, 2^{-13}, \ldots, 2^{15}]$ and $[k, max(2k, 50), max(4k, 100)]$, respectively. Here, $max(\cdot, \cdot)$ outputs the larger value of the two inputs. K-means clustering is adopted on the final representation $\mathbf{S}$ to assign an appropriate label for each sample. In the experiment, we repeat the clustering process for 50 times with random initialization and report the result with the smallest k-means distortion. We adopt the widely used clustering accuracy (ACC), normalized mutual information (NMI) and purity to evaluate the clustering performance. Specifically, the definition of the ACC is as follows,

$$ACC = \frac{\sum_{i=1}^{n} \delta(y_i, map(c_i))}{n}, \tag{15}$$

where c_i and y_i represent the obtained cluster label and the provided ground-truth label of $\mathbf{x}_i (1 \leq i \leq n)$, n is the number of samples, $\delta(u, v)$ is the delta function that equals to one if $u = v$ and equals zero otherwise, and $map(c_i)$ is the permutation mapping function that maps each cluster label $c_i$to the equivalent label from data. The best mapping can be found by using the Kuhn-Munkres algorithm [48]. Similarly, NMI is defined as follows. Let $\mathbf{y}$ and $\mathbf{c}$ denote the set of clusters obtained from the ground truth and a clustering algorithm, respectively. Their mutual information metric $\mathbf{MI}(\mathbf{y}, \mathbf{c})$ is defined as follows:

$$\mathbf{MI}(\mathbf{y}, \mathbf{c}) = \sum_{y_i \in \mathbf{y}, c_j \in \mathbf{c}} p(y_i, c_j) log_2 \frac{p(y_i, c_j)}{p(y_i)p(c_j)}, \tag{16}$$

where $p(y_i)$ and $p(c_j)$ are the probabilities that a sample arbitrarily selected from data belongs to the clusters y_i and c_j, respectively, and $p(y_i, c_j)$ is the joint probability that the arbitrarily selected samples belongs to the clusters y_i and c_j at the same time. The normalized mutual information (NMI) is then defined as follows:

$$\mathbf{NMI}(\mathbf{y}, \mathbf{c}) = \frac{\mathbf{MI}(\mathbf{y}, \mathbf{c})}{max(H(\mathbf{y}), H(\mathbf{c}))}, \tag{17}$$

where $H(\mathbf{y})$ and $H(\mathbf{c})$ are the entropies of $\mathbf{y}$ and $\mathbf{c}$, respectively.

In the following parts, we conduct comprehensive experiments to study the properties of CSA-MKC from five aspects: the advantage of joint sampling and multi-view information fusion, clustering performance, running time, parameter sensitivity and convergence. All our experiments are conducted on a desktop computer with a 3.6GHz Intel Core i7 CPU and 64GB RAM. The GPU version of our code is tested on a Titan XP GPU, with CUDA 10.0.130.

TABLE III: Ablation study. Clustering performance and time cost of the four designed algorithms are reported. Ablation1 is the algorithm which isolates sampling from MKC and uses a distinct sampling matrix for each view. Ablation2 is the algorithm which combines sampling with MKC and uses a distinct sampling matrix for each view. Ablation3 is the algorithm without a sampling operation. Our proposed algorithm is listed in the last column, it is an algorithm that integrates sampling with MKC while using a common sampling matrix for all views. In the computational time, the time consumption of the CPU version of all the algorithms are reported.

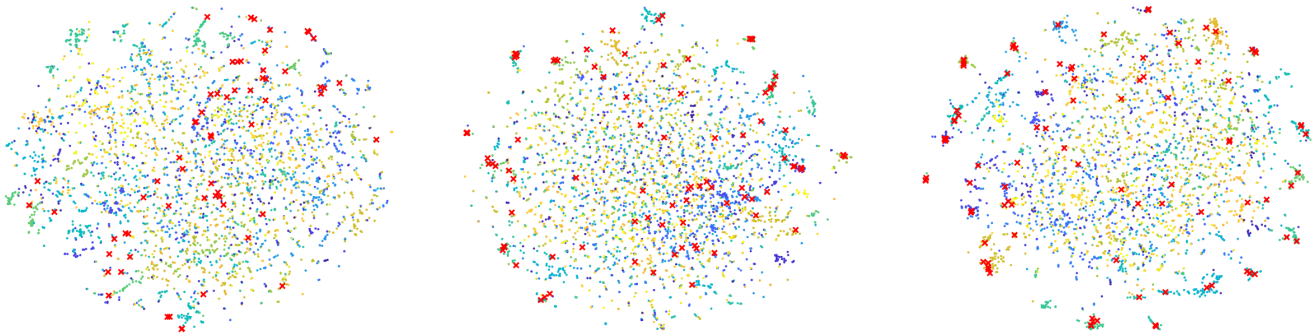
Datasets	Ablation1	Ablation2	Ablation3	Ablation4
ACC (%)				
Flower17	62.28	63.75	**69.78**	66.76
Plant	68.19	68.19	67.34	**70.43**
Flower102	37.06	40.47	42.76	**49.09**
CCV	26.46	31.85	31.45	**32.59**
Mfeat	96.70	86.60	96.85	**97.30**
NMI(%)				
Flower17	61.71	60.27	**68.02**	63.37
Plant	36.84	39.68	37.61	**39.69**
Flower102	52.48	54.67	59.84	**61.82**
CCV	21.25	26.46	26.04	**27.61**
Mfeat	92.47	87.54	93.04	**93.63**
Purity (%)				
Flower17	62.72	64.78	**71.32**	68.75
Plant	71.49	71.81	67.34	**70.43**
Flower102	42.31	47.20	48.60	**54.81**
CCV	29.59	34.86	33.35	**35.39**
Mfeat	96.70	87.50	96.85	**97.30**
Computational time (s)				
Flower17	**3.46**	5.48	15.52	4.20
Plant	**20.70**	33.80	48.25	17.48
Flower102	**56.34**	141.64	1027.1	170.85
CCV	**8.53**	40.46	866.52	60.10
Mfeat	**7.48**	21.17	68.07	13.79

B. Ablation Study

In our experiments, to illustrate the effectiveness of combining sampling with multiple kernel clustering, we design four algorithms and compare their performance. The first algorithm, denoted as Ablation1, is the version that isolates the process of sampling, and uses a distinct sampling matrix for each view. The second algorithm, denoted as Ablation2, is the version that combines sampling with MKC, but uses a distinct sampling matrix for each view. The third algorithm, denoted as Ablation3, is the version which has no sampling process, i.e., $\mathbf{P} = \mathbf{I}_n$. The fourth algorithm is our proposed algorithm which combines sampling with MKC and uses a consensus sampling matrix for all views.

1) Statistical Comparison: The clustering accuracy, NMI, purity and running time of the four compared algorithms are reported in Table III. From the table we have the following observations:

- The performance of Ablation2 and the proposed algorithm (Ablation4) is consistently better than that of Ablation1. It indicates that integrating sampling with MKC does improve the performance of clustering.



(a) Ablation1 (b) Ablation2 (c) Proposed

Fig. 1: Illustration of clustering results and sampling points on Flower102. In these figures, the colored dots represent the cluster structure of each algorithm generated by the t-SNE [49] algorithm, while the red crosses indicate the locations of the sampled points. Specifically, Figure (a), (b), and (c) are the representative results of Ablation1, Ablation2, and the proposed algorithm, respectively.

- As the Ablation4 further surpasses the Ablation2 for a consistent gap in terms of clustering performance, it is clear that learning a consensus sampling matrix which has fused information from different views is better for clustering performance improvement than generating sampling matrices independently.
- The sampling operation does improve the multi-view clustering speed. Specifically, conducting sampling and MVC separately (Ablation1), i.e., sample the anchor points first and then do clustering with the fixed anchor points, can speed up the classic MKC algorithm for more than 20 times on average (Ablation3). Conducting sampling and MVC simultaneously (Ablation2 and Ablation4) is slower than doing the two processes separately, however, they still improves the learning speed for **more than 7 times** against the classic algorithm.
- The sampling operation not only improves the multi-view clustering speed but also improve the clustering performance against the classic algorithm to some extent. Specially, comparing to the algorithm that adopt the whole dataset for sample reconstruction (Ablation3), the proposed algorithm achieves performance enhancement on four of the five datasets and improves the clustering ACC for 1.6% on average. The reason behind is similar to the mechanism of feature selection, i.e., selecting the discriminative and representative samples among datasets can to some extent get rid of the adverse effects of the redundant information and outliers.

2) Clustering Results and Anchor Points Illustration: To reveal the affect of merging the sampling and multi-view clustering processes, we first illustrate the clustering results of three compared algorithms, i.e., Ablation1, Ablation2 and our proposed algorithm (Ablation4) with t-SNE [49] algorithm and then marked the representative anchor points of these algorithms. In Fig.1, the colored dots represent the revealed distribution of sample points of the Flower102 dataset. Each different color indicates a different cluster. The red crosses indicate the learned anchor points. In Fig. 1, sub-figure (a) represents the result of conducting the sampling process independently with the MKC process. Sub-figure (b) represents the result of conducting the two processes simultaneously but using a specific sampling matrix for each view. Sub-figure (c) represents the results of conducting and two processes simultaneously and using a consensus sampling matrix for all views.

From the figures we have the observation that learning the sampling process together with the MKC tends to achieve better clustering performance. The preserved distribution of the proposed algorithm better suit the intrinsic cluster structure of data. Also, learning a consensus sampling matrix by integrating information from each view helps make the sampled anchor points to be more suitable for the revealing of the underlining overall geometric structure. The illustration in this part, shed light on the mechanism of the effectiveness of the proposed algorithm.

C. Comparison with State-of-the-art Algorithms

In this part we further compare our algorithm with seven state-of-the-art MKC algorithms to verify its effectiveness. The information of the compared algorithms are listed as follows.

- **Average multiple kernel k-means (A-MKKM)** Average multiple kernel k-means (A-MKKM) uniformly combines each kernel and uses the average kernel for clustering.
- **Single best kernel k-means (SB-KKM)** Single best kernel k-means algorithm conducts kernel k-means on each single kernel and reports the best result.
- **Multiple kernel k-means with matrix-induced regularization (MKKM-MR)** Multiple kernel k-means with matrix-induced regularization [3] introduces a diversity induction term to better merge multi-source information in MKC.
- **Multiple kernel clustering with local kernel alignment maximization (MKC-LKA)** Multiple kernel clustering with local kernel alignment maximization [4] try to better preserve the intrinsic local geometric structure among data by maximizing the local kernel alignment.
- **Multi-view clustering via late fusion alignment maximization (MVC-LFA)** In MVC via late fusion alignment

TABLE IV: Clustering performance comparison between the state-of-the-art algorithms. In this table, ACC, NMI, purity of different clustering algorithms on six popular datasets are reported. The red boldface indicates the best performance among all the compared algorithms.

Datasets	A-MKKM	SB-KKM	CRSC [16]	MKC-LKA [4]	MKKM-MR [3]	MVC-LFA [36]	LMVSC [34]	proposed
ACC (%)								
Flower17	51.03	42.06	51.76	60.69	59.69	61.16	62.28	**66.76**
Plant	61.70	51.60	61.91	52.34	62.87	61.81	68.19	**70.43**
Flower102	27.29	33.13	38.00	40.84	40.24	42.16	37.06	**49.09**
CCV	19.74	20.08	18.01	23.49	22.47	27.56	26.46	**32.59**
Mfeat	95.20	86.00	83.30	96.25	94.65	95.30	96.70	**97.30**
MNIST	77.33	77.89	-	-	-	78.58	82.85	**87.17**
NMI(%)								
Flower17	50.19	45.14	53.19	57.27	57.11	60.79	61.71	**63.37**
Plant	26.82	17.18	25.83	21.35	28.29	26.87	36.84	**39.69**
Flower102	46.32	48.99	54.95	57.60	57.27	60.48	52.48	**61.82**
CCV	17.16	17.73	18.89	17.11	18.62	20.59	21.25	**27.66**
Mfeat	89.83	75.79	76.48	91.90	89.04	90.02	92.74	**93.63**
MNIST	74.28	76.50	-	-	-	75.47	69.87	**77.44**
Purity (%)								
Flower17	51.99	44.63	53.68	61.79	60.03	62.32	62.72	**68.75**
Plant	61.70	56.38	62.45	58.19	62.87	61.81	69.49	**70.43**
Flower102	32.28	38.74	45.04	48.21	46.39	50.44	42.31	**54.81**
CCV	23.98	23.48	26.80	22.93	25.69	30.71	29.59	**35.39**
Mfeat	95.20	86.00	83.30	96.25	94.65	95.30	96.70	**97.30**
MNIST	81.53	82.63	-	-	-	82.65	82.86	**87.19**

TABLE V: Clustering speed comparison between the state-of-the-art algorithms. In this table, the average CPU clustering time consumption of different clustering algorithms on six popular datasets are reported.

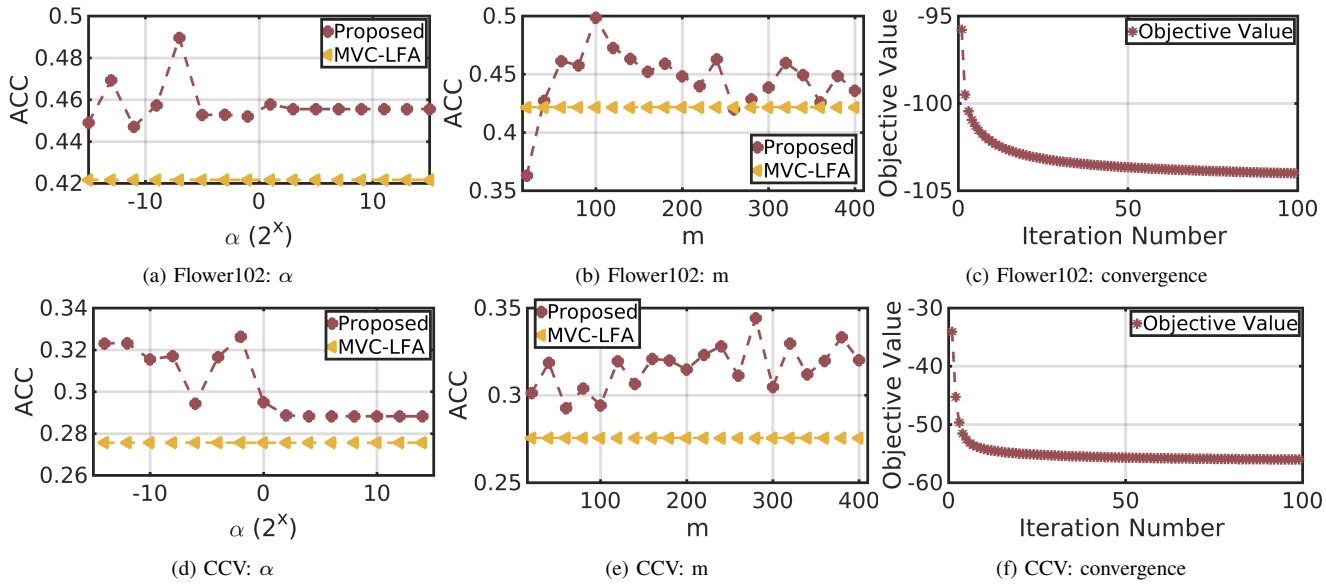
Datasets	A-MKKM	SB-KKM	CRSC [16]	MKC-LKA [4]	MKKM-MR [3]	MVC-LFA [36]	LMVSC [34]	proposed
Computational time (s)								
Flower17	0.74	4.58	10.04	8.15	5.47	2.88	20.02	4.20
Plant	0.16	8.15	129.20	31.95	5.69	2.33	122.91	17.48
Flower102	27.51	120.50	426.60	427.40	382.13	97.59	233.06	170.85
CCV	6.98	17.84	283.96	281.92	161.20	30.08	47.55	60.10
Mfeat	0.91	10.70	147.65	13.93	5.95	4.57	48.72	13.79
MNIST	26.19	95.72	-	-	-	281.01	433.50	1142.05
Average	7.26	51.49	199.49	152.67	112.09	27.49	94.45	53.28

maximization (MVC-LFA) [36], the authors maximally align the consensus partition with the weighted base partitions for efficient clustering.

- **Large-scale multi-view subspace clustering (LMVSC)** In large-scale multi-view subspace clustering, Kang et.al [34] take a sampling strategy for efficient multi-view subspace clustering.
- **Compressed subspace alignment-based multiple kernel clustering (CSA-MKC)** Compressed subspace alignment-based multiple kernel clustering is our proposed algorithm which integrate sampling with MKC to have the sampling process better serve the multi-view clustering process.

1) Clustering Performance Comparison: The clustering results of the compared state-of-the-art algorithms are reported in Table IV. In this table, since the MNIST dataset with 60,000 samples is too large for the memory inefficient early fusion multiple kernel clustering algorithms and the corresponding algorithms run out of memory on that dataset, we omit the results on the MNIST dataset of these algorithms. From the results in Table IV, we have the following observations. First, average kernel and single best kernel methods are strong competitors against other multiple kernel clustering algorithms that they perform well on most of the compared datasets. Second, the recently proposed MKC algorithms are effective in improving the clustering performance that they surpass the A-MKKM and SB-KKM algorithm in most of the circumstances. Third, our algorithm achieves the best performance on all six datasets, and improves the second best algorithm for more than 5% on the criteria of ACC on the flower102 and CCV datasets. Specifically, the two algorithms enhanced with sampling strategies (i.e., LMVSC and the proposed algorithm) keep achieve at least comparable performance with the compared algorithms which adopt the whole dataset for learning, indicating that with careful design, the sampling operation can speed up the learning process and in the mean time achieve good clustering performance. Moreover, our algorithm outperforms the LMVSC with a considerable gap, validating the effectiveness of our simultaneous sampling and learning mechanism.

2) Computational Time Comparison: We also report the computational cost of the compared state-of-the-art algorithms in Figure 3. In this table, as the CRSC, MKC-LKA and



(a) Flower102: α (b) Flower102: m (c) Flower102: convergence

(d) CCV: α (e) CCV: m (f) CCV: convergence

Fig. 2: Parameter sensitivity and convergence of the proposed algorithm on the Flower102 and CCV dataset. In the first and the second columns are the parameter sensitivity of α and m, respectively. The last column is the convergence illustration of the proposed algorithm.

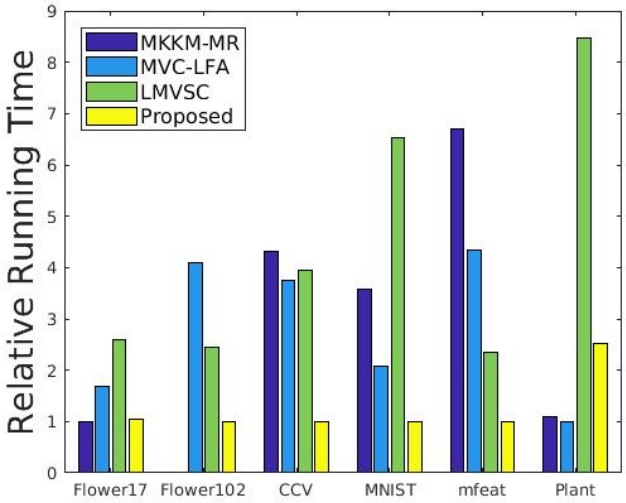


Fig. 3: The accelerated running time comparison. In this figure, the four fastest compared algorithms, i.e., MKKM-MR, MVC-LFA, LMVSC, and the proposed algorithm are further accelerated with the GPU and multi-core techniques. The corresponding relative computational time is illustrated. The taller a bar, is the larger the relative computational time will be.

MKKM-MR algorithm run out of memory on the MNIST dataset, we omit the computational time of the corresponding algorithms on the dataset. For the ease of comparison, we also report the average computational time of all the compared algorithms on the first five datasets with relatively smaller size. From the table we can find that the empirical results are consistent with the theoretical analysis in the discussion. Specifically, as a $\mathcal{O}(n^2)$ complexity algorithm, our proposed algorithm largely surpasses the $\mathcal{O}(n^3)$ algorithms, i.e., CRSC, MKC-LKA, and MKKM-MR. However, it is also much slower than the linear complexity algorithm, especially on the large scale MNIST dataset.

3) GPU and Multi-Core Accelerated Computational Time Comparison: To further speed up the compared algorithms and check their acceleration rates, we introduce two modifications to the implementation of the four fastest MKC algorithms, i.e., MKKM-MR, MVC-LFA, LMVSC and our proposed algorithm. The first modification is that we introduce multi-core parallel computation to the independent *for* iterations. The second modification is that we adopt the GPU acceleration package to further speed up the optimization procedure.

The relative computational time of the compared algorithms are illustrated in Figure 3. Specially, for each dataset, the relative time consumption of each dataset is calculated by dividing the time cost with the smallest time cost on the corresponding dataset. From the results, we find that the computational speed of all the compared algorithms are increased to different extent. Specifically, the acceleration rate of MKKM-MR, MVC-LFA, LMVSC, and the proposed algorithms are 3.66, 1.23, 3.29, and 15.1, respectively. To the MVC-LFA algorithm, as the bottleneck of the algorithm mainly lies with eigenvalue decomposition which is hard to parallel, it achieves the smallest acceleration rate. To the proposed algorithm, since more than 90% of the computational cost is spent on matrix manipulation, which is easy and suitable for parallelization, it achieves the largest acceleration rate. Moreover, the results in Figure 3 also shows that, although the proposed algorithm is with a larger computational complexity compared with the linear algorithms, it can achieve comparable efficiency with the help of multi-core and GPU acceleration. This setting further improves the scalability of the proposed algorithm.

D. Parameter Sensitivity and Convergence

Parameter Sensitivity. The proposed algorithm introduces two hyper-parameters, i.e., the number of anchor points m and the balancing coefficient α. To test the sensitivity of the proposed algorithm against these two parameters, we fix one parameter and tune the other in a large range. The comparison between the proposed algorithm with the second best algorithm on two representative datasets, i.e., Flower102 and CCV are illustrated in Fig.2(a,b,d,e). From these figures, we observe that: i) tuning both m and α are effective in improving the algorithm performance; ii) the algorithm is stable against the two parameters and good performance can be achieved with only a small number of anchor points; iii) the performance of our algorithm significantly surpasses the second best algorithm in most of the circumstances.

Algorithm Convergence. The objective value variation of our method on the Flower102 and CCV datasets are shown in Fig.2(c). As observed from this figure, the convergence curve decreases monotonically and quickly converges to a local minimal.

V. Conclusion

This paper proposes multiple kernel clustering with compressed subspace alignment (MKC-CSA) to improve both the clustering efficiency and accuracy in multiple kernel learning scenario. In our method, by modeling the sampling process with a linear compressing matrix, we merge sampling and MKC together in a unified framework and optimize the two tasks iteratively. As a consequence, the computational efficiency is largely improved and the clustering performance is considerably enhanced. In the future, we plan to extend our algorithm to a more general framework and use it as a platform to revisit existing multi-view clustering algorithms.

Acknowledgment

This work was supported in part by the National Key R&D Program of China 2018YFB1003203, and in part by the National Science Foundation of China under Grant 61773392, 61672528 and 62006237.

References

[1] B. Zhao, J. T. Kwok, and C. Zhang, "Multiple kernel clustering," in *Proceedings of the 2009 SIAM International Conference on Data Mining*. SIAM, 2009, pp. 638–649.

[2] J. Zhuang, J. Wang, S. C. Hoi, and X. Lan, "Unsupervised multiple kernel learning," 2011.

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

[4] 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.

[5] X. Liu, X. Zhu, M. Li, L. Wang, E. Zhu, T. Liu, M. Kloft, D. Shen, J. Yin, and W. Gao, "Multiple kernel kk-means with incomplete kernels," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 5, pp. 1191–1204, 2020.

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

[7] X. Guo, "Robust subspace segmentation by simultaneously learning data representations and their affinity matrix," in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[8] P. Zhou, L. Du, L. Shi, H. Wang, and Y.-D. Shen, "Recovery of corrupted multiple kernels for clustering," in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[9] X. Liu, S. Zhou, Y. Wang, M. Li, Y. Dou, E. Zhu, and J. Yin, "Optimal neighborhood kernel clustering with multiple kernels," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

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

[11] Y. Liang, D. Huang, and C.-D. Wang, "Consistency meets inconsistency: A unified graph learning framework for multi-view clustering," in *Proceedings of the IEEE International Conference on Data Mining*, 2019.

[12] S. Zhou, X. Liu, M. Li, E. Zhu, 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.

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

[14] 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 *The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20), New York, USA*, 2020.

[15] A. Kumar and H. Daume, "A co-training approach for multi-view spectral clustering," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011, pp. 393–400.

[16] A. Kumar, P. Rai, and H. Daume, "Co-regularized multi-view spectral clustering," in *NIPS*, 2011, pp. 1413–1421.

[17] G.-Y. Zhang, C.-D. Wang, D. Huang, W.-S. Zheng, and Y.-R. Zhou, "Tw-co-k-means: two-level weighted collaborative k-means for multi-view clustering," *Knowledge-Based Systems*, vol. 150, pp. 127–138, 2018.

[18] L. Huang, H.-Y. Chao, and C.-D. Wang, "Multi-view intact space clustering," *Pattern Recognition*, vol. 86, pp. 344–353, 2019.

[19] E. V. Strobl, K. Zhang, and S. Visweswaran, "Approximate kernel-based conditional independence tests for fast non-parametric causal discovery," *Journal of Causal Inference*, vol. 7, no. 1, 2019.

[20] Y. Yao, Y. Li, B. Jiang, and H. Chen, "Multiple kernel k-means clustering by selecting representative kernels," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–14, 2020.

[21] X. Peng, J. Feng, J. T. Zhou, Y. Lei, and S. Yan, "Deep subspace clustering," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–13, 2020.

[22] L. Wu and Y. Wang, "Robust hashing for multi-view data: Jointly learning low-rank kernelized similarity consensus and hash functions," *Image and Vision Computing*, vol. 57, pp. 58–66, 2017.

[23] Z. Zhang, L. Liu, F. Shen, H. T. Shen, and L. Shao, "Binary multi-view clustering," *IEEE TPAMI*, vol. 41, no. 7, pp. 1774–1782, 2018.

[24] X. Shen, W. Liu, I. Tsang, F. Shen, and Q.-S. Sun, "Compressed k-means for large-scale clustering," in *AAAI*, 2017.

[25] Z. Zhang, L. Liu, J. Qin, F. Zhu, F. Shen, Y. Xu, L. Shao, and H. Tao Shen, "Highly-economized multi-view binary compression for scalable image clustering," in *ECCV*, 2018, pp. 717–732.

[26] D. Huang, C.-D. Wang, J. Wu, J.-H. Lai, and C. K. Kwoh, "Ultra-scalable spectral clustering and ensemble clustering," *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[27] M. Hu and S. Chen, "One-pass incomplete multi-view clustering," *arXiv preprint arXiv:1903.00637*, 2019.

[28] P. Zhou, Y.-D. Shen, L. Du, F. Ye, and X. Li, "Incremental multi-view spectral clustering," *Knowledge-Based Systems*, vol. 174, pp. 73–86, 2019.

[29] L. Huang, C. D. Wang, H. Y. Chao, and P. S. Yu, "Mvstream: Multiview data stream clustering," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 9, pp. 3482–3496, 2020.

[30] Y. Li, F. Nie, H. Huang, and J. Huang, "Large-scale multi-view spectral clustering via bipartite graph," in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[31] D. Cai and X. Chen, "Large scale spectral clustering via landmark-based sparse representation," *IEEE transactions on cybernetics*, vol. 45, no. 8, pp. 1669–1680, 2014.

[32] P. Zhou, L. Du, X. Liu, Y. D. Shen, M. Fan, and X. Li, "Self-paced clustering ensemble," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2020.

[33] Z. Ren and Q. Sun, "Simultaneous global and local graph structure preserving for multiple kernel clustering," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–13, 2020.

[34] Z. Kang, W. Zhou, Z. Zhao, J. Shao, M. Han, and Z. Xu, "Large-scale multi-view subspace clustering in linear time," *arXiv preprint arXiv:1911.09290*, 2019.
[35] 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 TPAMI*, 2018.
[36] 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*. AAAI Press, 2019, pp. 3778–3784.
[37] X. Liu, M. Li, C. Tang, J. Xia, J. Xiong, L. Liu, M. Kloft, and E. Zhu, "Efficient and effective regularized incomplete multi-view clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020.
[38] L. Parsons, E. Haque, and H. Liu, "Subspace clustering for high dimensional data: a review," *Acm Sigkdd Explorations Newsletter*, vol. 6, no. 1, pp. 90–105, 2004.
[39] X. Peng, H. Tang, L. Zhang, Z. Yi, and S. Xiao, "A unified framework for representation-based subspace clustering of out-of-sample and large-scale data," *IEEE TNNLS*, vol. 27, no. 12, pp. 2499–2512, 2015.
[40] X. Peng, Z. Yu, Z. Yi, and H. Tang, "Constructing the l2-graph for robust subspace learning and subspace clustering," *IEEE transactions on cybernetics*, vol. 47, no. 4, pp. 1053–1066, 2016.
[41] C. Cortes, M. Mohri, and A. Rostamizadeh, "Algorithms for learning kernels based on centered alignment," *Journal of Machine Learning Research*, vol. 13, no. Mar, pp. 795–828, 2012.
[42] Y. Wang, X. Lin, L. Wu, W. Zhang, Q. Zhang, and X. Huang, "Robust subspace clustering for multi-view data by exploiting correlation consensus," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3939–3949, 2015.
[43] F. Nie, J. Yuan, and H. Huang, "Optimal mean robust principal component analysis," in *International conference on machine learning*, 2014, pp. 1062–1070.
[44] T. Wang, S. Tian, H. Huang, and D. Deng, "Learning by local kernel polarization," *Neurocomputing*, vol. 72, no. 13-15, pp. 3077–3084, 2009.
[45] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
[46] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.
[47] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
[48] L. Lovász and M. D. Plummer, *Matching theory*. American Mathematical Soc., 2009, vol. 367.
[49] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

Sihang Zhou received his PhD degree from National University of Defense Technology (NUDT), China. He is now lecturer at College of Intelligence Science and Technology, NUDT. His current research interests include machine learning and medical image analysis. Dr. Zhou has published 20+ peer-reviewed papers, including IEEE T-IP, IEEE T-NNLS, IEEE T-MI, Information Fusion, Medical Image Analysis, AAAI, MICCAI, etc.

Qiyuan Ou received the B.S. degree in network engineering from the National University of Defense Technology (NUDT), China, in 2018, where she is currently pursuing the M.S. degree in CS department with the PRMI group. Her research interests include multiple kernel clustering.

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

Siqi Wang received the BS degree and the PhD degree in computer science and technology from the National University of Defense Technology, China. He is currently an assistant professor with the State Key Laboratory of High Performance Computing (HPCL), National University of Defense Technology , China. His main research include anomaly / outlier detection, pattern recognition and unsupervised learning. His works have been published on leading conferences and journals, such as, NeurIPS, AAAI, IJCAI, ACM MM, ICPR, Pattern Recognition, IEEE Transactions on Cybernetics and Neurocomputing. He also serves as a reviewer for several international journals, including the IEEE Transactions on Cybernetics, the IEEE Transactions on Automation Science and Engineering, Artificial Intelligence Review, and International Journal of Machine Learning and Cybernetics.

Luyan Liu has received her PhD degree from Shanghai Jiao Tong University (SJTU). She is now a senior researcher in Tencent Jarvis Lab. Her current research interests include medical image segmentation, 2D/3D instance segmentation, NAS and model generalization. Dr. Liu has published 10 peer-reviewed papers, including MICCAI, IEEE T-MI, IJCAI, etc.

Siwei Wang is a graduate student in National University of Defense Technology (NUDT), China. His current research interests include kernel learning, unsupervised multiple-view learning, scalable clustering and deep unsupervised learning.

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

Jianping Yin Jianping Yin received the Ph.D. degree from the National University of Defense Technology (NUDT), Changsha, China. He is currently a Distinguished Professor with the Dongguan University of Technology, Dongguan, China. His current research interests include pattern recognition and machine learning. He has published over 150 peer-reviewed papers in journals and conferences, including the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, the IEEE TRANSACTIONS ON N EURAL NETWORKS AND LEARNING SYSTEMS, PR, the Advertising Agencies Association of India, and the International Joint Conferences on Artificial Intelligence. Dr. Yin was a recipient of the China National Excellence Doctoral Dissertation' Supervisor Award and the National Excellence Teacher Award. He served on the Technical Program Committees of over 30 international conferences and workshops.

Xin Xu (M'07-SM'12) received the B.S. degree in electrical engineering from the Department of Automatic Control, National University of Defense Technology (NUDT), Changsha, China, in 1996, and the Ph.D. degree in control science and engineering from the College of Mechatronics and Automation, NUDT, in 2002. He has been a Visiting Professor with The Hong Kong Polytechnic University, the University of Alberta, the University of Guelph, and the University of Strathclyde, U.K. He is currently a Full Professor with the Institute of Unmanned Systems, College of Intelligence Science and Technology, NUDT. He has co-authored more than 160 papers in international journals and conferences and four books. His research interests include intelligent control, reinforcement learning, approximate dynamic programming, machine learning, robotics, and autonomous vehicles. He is a member of the IEEE CIS Technical Committee on Approximate Dynamic Programming and Reinforcement Learning and the IEEE RAS Technical Committee on Robot Learning. He received the National Science Fund for Outstanding Youth in China and the second-class National Natural Science Award of China. He has served as an Associate Editor or Guest Editor for Information Sciences, International Journal of Robotics and Automation, IEEE Transactions on Systems, Man, and Cybernetics: Systems, Intelligent Automation and Soft Computing, the International Journal of Adaptive Control and Signal Processing and Acta Automatica Sinica.