

Incremental Nyström-based Multiple Kernel Clustering

Yu Feng¹, Weixuan Liang¹, Xinhang Wan¹, Jiyuan Liu², Suyuan Liu¹, Qian Qu¹, Renxiang Guan¹, Huiying Xu^{3*}, Xinwang Liu^{1*}

¹College of Computer Science and Technology, National University of Defense Technology, Changsha, China, 410073

²College of Systems Engineering, National University of Defense Technology, Changsha, China, 410073

³School of Computer Science and Technology, Zhejiang Normal University, Jinhua, China, 321004

{fengyu, xinwangliu}@nudt.edu.cn

Abstract

Existing Multiple Kernel Clustering (MKC) algorithms commonly utilize the Nyström method to handle large-scale datasets. However, most of them employ uniform sampling for kernel matrix approximation, hence failing to accurately capture the underlying data structure, leading to large approximation errors. Additionally, they often use the same landmark points for all kernel matrix approximations, reducing kernel diversity. Moreover, in scenarios where approximate kernel matrices emerge over time, these methods require storing historical kernel information and recalculating, resulting in inefficient resource utilization. To address these issues, we propose a novel MKC algorithm, termed Incremental Nyström-based Multiple Kernel Clustering (INMKC). Specifically, leverage score sampling is utilized to reduce kernel approximation errors and enhance kernel diversity. Furthermore, we employ a consensus clustering structure that aligns with the newly emerged base kernel matrix for updates, avoiding recalculating previous kernel matrices, thus saving substantial computational resources. Additionally, we tackle the challenge of aligning incremental approximate kernels with different landmark points. Extensive experiments on the proposed INMKC demonstrate its effectiveness and efficiency compared to state-of-the-art methods.

Introduction

Kernel learning (Hofmann, Schölkopf, and Smola 2008) is essential in machine learning, allowing classical algorithms to solve nonlinear problems by mapping data into high-dimensional Hilbert space. Specifically, kernel k -means can effectively partition linearly inseparable datasets. However, the choice of kernel functions or kernel matrices derived from various data views can significantly impact clustering performance (Chao, Sun, and Bi 2021). To address this, Multiple Kernel Clustering (MKC) (Zhao, Kwok, and Zhang 2009) constructs a set of base kernel matrices in advance and fuses them linearly to obtain an optimal kernel matrix. Based on this principle, numerous MKC algorithms have been developed (Huang, Chuang, and Chen 2011; Du et al. 2015; Liu 2023; Yang et al. 2023; Wan et al. 2024a).

Although these methods improve clustering performance, they struggle with large-scale and dynamic datasets. MKC

has high computational complexity. Specifically, storing base kernel matrices requires $\mathcal{O}(mn^2)$ space complexity and decomposing them takes $\mathcal{O}(n^3)$ time complexity, where n and m are the number of samples and base kernel matrices, respectively. To enable MKC to handle large-scale datasets, most existing research (Lu et al. 2022; Liang et al. 2023, 2024b) uses the Nyström method based on uniform sampling to approximate kernel matrices. However, uniform sampling is inadequate for datasets with unbalanced cluster distributions, often leading to information redundancy or neglect of small clusters, and failing to accurately capture the structure of the dataset (Musco and Musco 2017). Moreover, using the same landmark points for constructing all base kernel matrices limits the ability to capture complementary information across different kernels (Yang and Wang 2018).

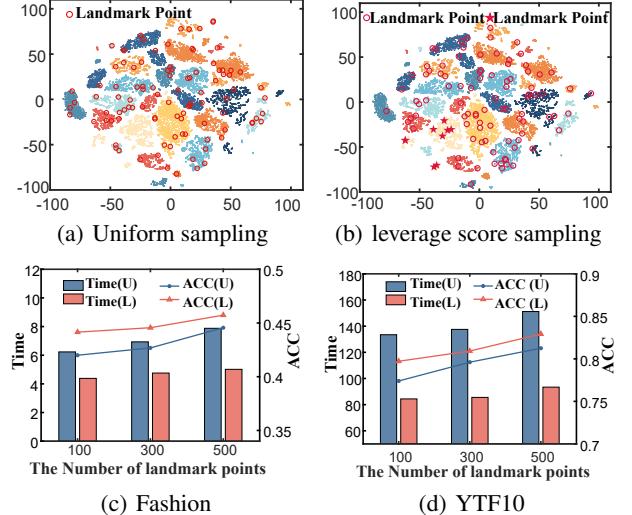


Figure 1: (a) and (b): Landmarks obtained from uniform and leverage score sampling on YTF10 (dataset with unbalanced cluster distributions). (c) and (d): The variation in Accuracy (ACC) and running time with different numbers of landmark points and different sampling methods.

Leverage score sampling, by contrast, selects landmark points based on their statistical significance in representing the dataset. This approach improves kernel accuracy and

*Corresponding author

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

diversity by focusing on the most influential points, particularly where uniform sampling might miss smaller, crucial clusters (Musco and Musco 2017). To better illustrate our motivation, we conducted two comparison experiments. First, we used t-SNE (van der Maaten and Hinton 2008) to visualize the distributions of landmarks sampled by both uniform sampling and leverage score sampling methods. As illustrated in Figures 1 (a) and (b), uniform sampling tends to oversample dense regions in datasets with unbalanced cluster distributions, whereas leverage score sampling covers the entire dataset more efficiently. In the second experiment, we selected landmark points using both sampling methods and applied the approximation spectral clustering algorithm from (Chen and Cai 2011) on the training set. The corresponding clustering accuracy with different numbers of landmarks and different sampling methods is shown in Figure 1 (c) and (d). The results demonstrate that leverage score sampling improves clustering performance when the number of landmark points is the same. Therefore, we propose using leverage score sampling to approximate kernel matrices, enhancing both the accuracy of the approximation and the diversity among the base kernel matrices.

Despite addressing the issue of kernel matrix approximation, another significant limitation of MKC lies in its inefficiency in handling scenarios where base kernel matrices are accumulated over time. Specifically, in hazardous gas detection systems (Vergara et al. 2013), multiple sensors periodically collect data on various gases. Each data collection represents a view, resulting in a corresponding base kernel matrix, which increases over time. A straightforward approach involves storing historical and incoming matrices and applying existing MKC methods to the entire set for clustering. However, this approach leads to significant waste of time and storage space, as well as the underutilization of previously acquired model knowledge. Incremental learning (Wang et al. 2023) offers a solution by facilitating continuous learning from dynamically distributed data, allowing knowledge to be updated incrementally without retraining on all previous data. While widely applied in supervised learning, recent studies (Zhou et al. 2019; Yin et al. 2021; Wan et al. 2022, 2024b; Liang et al. 2024a; Hu and Chen 2019; Cai et al. 2023; Qu et al. 2024) have begun integrating incremental learning into unsupervised clustering algorithms. However, research on incrementally handling base kernel matrices is limited, especially with varying landmark points. Misalignment can lead to significant errors and inconsistent clustering, particularly in incremental learning where new data adds complexity. Addressing this is crucial to maintaining accuracy and robustness.

To address the above issues, we propose a novel algorithm, termed Incremental Nyström-based Multiple Kernel Clustering (INMKC). This method employs leverage score sampling to approximate the base kernel matrix and integrates matrix factorization with incremental learning into a unified framework, effectively tackling the large-scale MKC problem with incremental base kernels. The specific framework of INMKC is shown in Figure 2. As illustrated, only the consensus cluster structure matrix \mathbf{H}_p^* and the latest data view \mathbf{X}_p need to be stored. Additionally, only the latest ap-

proximate base kernel matrix \mathbf{G}_p needs to be processed during the iteration. Thus, the proposed INMKC significantly reduces computational complexity. The main contributions of the paper can be summarized as follows:

- We employ leverage score sampling for accurate kernel approximation and introduce a novel method for aligning and fusing approximate kernel matrices.
- INMKC is the first to address large-scale base kernel increment issues by integrating matrix decomposition with incremental learning, enabling the aligned fusion of approximate kernels with different landmark points.
- To solve the optimization problem, we develop a four-step alternating strategy and prove its convergence both theoretically and experimentally. Extensive experiments demonstrate the effectiveness and efficiency of the proposed INMKC compared to state-of-the-art methods.

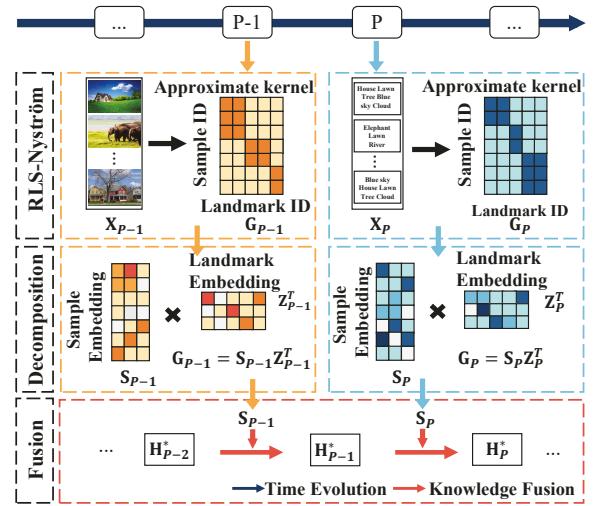


Figure 2: Framework of the proposed INMKC method.

Related Work

In this section, we will briefly overview related research on scalable multiple kernel clustering, the ridge leverage scores Nyström method, and incremental learning.

Scalable Multiple Kernel Clustering

High complexity is a critical issue for MKC algorithms. The method described in (Liang et al. 2024b) extends SimpleMKM to large-scale datasets by uniformly sampling s landmark points and constructing the approximate base kernel matrices $\{\mathbf{M}_p \in \mathbb{R}^{n \times s}\}_{p=1}^m$ between the n sample points and the s landmark points. The objective function is formulated as follows:

$$\begin{aligned} & \min_{\gamma} \max_{\mathbf{U}, \mathbf{V}} \frac{1}{\sqrt{ns}} \text{Tr}(\mathbf{U}^\top \mathbf{M}_\gamma \mathbf{V}), \\ & \text{s.t. } \mathbf{U}^\top \mathbf{U} = \mathbf{I}_k, \mathbf{V}^\top \mathbf{V} = \mathbf{I}_k, \end{aligned} \quad (1)$$

where $\mathbf{M}_\gamma = \sum_{p=1}^m \gamma_p^2 \mathbf{M}_p$, $\mathbf{U} \in \mathbb{R}^{n \times k}$ and $\mathbf{V} \in \mathbb{R}^{s \times k}$. However, this method has two critical drawbacks: 1) It uses

uniform sampling to approximate the kernel matrix, which affects the accuracy of the approximation. Additionally, the proposed fusion method can only handle matrices with the same landmark points and cannot fuse matrices with different landmark points. 2) Over time, when a new base kernel matrix emerges, this method requires fusing all previous base kernel matrices before clustering, leading to resource waste.

Ridge Leverage Scores Nyström Method

Leverage score sampling (Alaoui and Mahoney 2015) involves computing the ridge leverage score for each data point x_i , which is then used as the sampling probability for selecting landmark points. The ridge leverage score, with respect to the kernel matrix \mathbf{K} , refers to the λ -ridge leverage score for any $\lambda > 0$ as follows,

$$l_i^\lambda(\mathbf{K}) \stackrel{\text{def}}{=} (\mathbf{K}(\mathbf{K} + \lambda\mathbf{I})^{-1})_{i,i}, \quad (2)$$

where \mathbf{I} is an $n \times n$ identity matrix. Musco proposed the recursive RLS-Nyström algorithm (Musco and Musco 2017), which requires only $\mathcal{O}(ns)$ kernel evaluations and $\mathcal{O}(ns^2)$ computation time. We will use this algorithm to approximate the base kernel matrices, achieving linear complexity.

Incremental Learning

Traditional machine learning models are often designed to capture static data distributions, while incremental learning is adapted to learn from dynamic data distributions. To address domain incremental learning, which can be viewed as the increment of views in multi-view clustering(MVC), (Wan et al. 2022) maintains a partition matrix and updates knowledge using new data views. Upon receiving the partition matrix \mathbf{H}_t at time t , the final consensus partition matrix \mathbf{H}_t^* can be obtained by maximizing the following objective function,

$$\begin{aligned} & \max_{\tilde{\mathbf{H}}_t, \mathbf{W}_t} \text{Tr}(\tilde{\mathbf{H}}_t^\top \mathbf{H}_t \mathbf{W}_t) + \lambda \text{Tr}(\tilde{\mathbf{H}}_t^\top \mathbf{H}_{t-1}^*), \\ & \text{s.t. } \tilde{\mathbf{H}}_t^\top \tilde{\mathbf{H}}_t = \mathbf{I}_k, \mathbf{W}_t^\top \mathbf{W}_t = \mathbf{I}_k, \end{aligned} \quad (3)$$

where \mathbf{W}_t is the permutation matrix. However, this method overlooks the computational complexity involved in constructing the consensus partition and cannot directly handle base kernel matrices.

Methodology

In this section, we first give the notations and problem definition, then outline the objective formulation of INMKC. We proceed by proposing a four-step alternating optimization algorithm, followed by a discussion on its convergence and complexity.

Notation and Problem statement

Vectors and matrices are denoted by lowercase bold and uppercase bold letters, respectively.

Our method addresses the problem of large-scale clustering for data views that increase over time. Let \mathbf{X}_p represent the data view at time p , which is transformed into a

base kernel matrix \mathbf{K}_p using a Gaussian kernel function. As new views emerge, the corresponding base kernel matrices accumulate. To tackle this, we employ the recursive RLS-Nyström algorithm (Musco and Musco 2017) to obtain an approximate kernel matrix \mathbf{G}_p at time p . The matrix factorization of \mathbf{G}_p reveals the clustering structure at the current time, which is then integrated with the previous consensus cluster structure \mathbf{H}_{p-1}^* using incremental learning.

The Proposed Formulation

We first construct a basic model for \mathbf{G}_p to efficiently explore its underlying clustering structure and facilitate knowledge fusion. According to Theorem 4.1 and Theorem 4.2 in (Liang et al. 2024c), the rank of the expected kernel matrix is approximately k , and the empirical kernel matrix is also close to the expected one. Theorem 8 in (Musco and Musco 2017) further supports that the approximated kernel matrix \mathbf{G}_p is close to the empirical kernel matrix. Thus, we can infer that $\text{rank}(\mathbf{G}_p) = k$. We then decompose \mathbf{G}_p to obtain the embedding factors of the sample points in the k -dimensional subspace at time p , representing the cluster structure.

$$\begin{aligned} & \min_{\mathbf{S}_p, \mathbf{Z}_p} \|\mathbf{G}_p - \mathbf{S}_p \mathbf{Z}_p^\top\|_F^2, \\ & \text{s.t. } \mathbf{S}_p^\top \mathbf{S}_p = \mathbf{I}_k, \mathbf{Z}_p^\top \mathbf{Z}_p = \mathbf{I}_k, \end{aligned} \quad (4)$$

where $\mathbf{S}_p \in \mathbb{R}^{n \times k}$ and $\mathbf{Z}_p \in \mathbb{R}^{s \times k}$ respectively represent the embeddings of the sample points and landmark points, encoding their underlying information. Following (Han and Kim 2015), we apply only the orthogonal constraint to these matrices, omitting the non-negativity constraint. This enhances the diversity of \mathbf{S}_p and \mathbf{Z}_p , facilitating the fusion of \mathbf{S}_p . We consider introducing incremental learning to fuse the cluster structure at time p with the consensus cluster structure at time $p-1$. This approach allows us to maintain only the latest consensus structure, avoiding the need to store all base kernel matrices. The objective function is as follows:

$$\begin{aligned} & \min_{\mathbf{H}_p^*, \mathbf{Q}_p} \|\mathbf{H}_p^* - \mathbf{S}_p \mathbf{Q}_p\|_F^2 + \|\mathbf{H}_p^* - \mathbf{H}_{p-1}^*\|_F^2, \\ & \text{s.t. } (\mathbf{H}_p^*)^\top \mathbf{H}_p^* = \mathbf{I}_k, \mathbf{Q}_p^\top \mathbf{Q}_p = \mathbf{I}_k, \end{aligned} \quad (5)$$

where $\mathbf{H}_p^* \in \mathbb{R}^{n \times k}$ represents the consensus cluster structure at time p . The first term in Eq.(5) aligns the consensus cluster structure \mathbf{H}_p^* with the cluster structure \mathbf{S}_p at time p . The matrix $\mathbf{Q}_p \in \mathbb{R}^{k \times k}$ denotes the matching matrix. The second term integrates the current consensus cluster structure \mathbf{H}_p^* with the historical knowledge \mathbf{H}_{p-1}^* .

To efficiently learn the consensus cluster structure \mathbf{H}_p^* and the embedding matrix \mathbf{S}_p of the base kernel matrix \mathbf{G}_p , we integrate matrix decomposition with incremental learning into a unified framework. This leads to the final objective function:

$$\begin{aligned} & \min_{\mathbf{x}} \|\mathbf{G}_p - \mathbf{S}_p \mathbf{Z}_p^\top\|_F^2 + \|\mathbf{H}_p^* - \mathbf{S}_p \mathbf{Q}_p\|_F^2 + \|\mathbf{H}_p^* - \mathbf{H}_{p-1}^*\|_F^2, \\ & \text{s.t. } \mathbf{S}_p^\top \mathbf{S}_p = \mathbf{I}_k, \mathbf{Z}_p^\top \mathbf{Z}_p = \mathbf{I}_k, (\mathbf{H}_p^*)^\top \mathbf{H}_p^* = \mathbf{I}_k, \mathbf{Q}_p^\top \mathbf{Q}_p = \mathbf{I}_k, \end{aligned} \quad (6)$$

for better clarity, we introduce decision variable \mathcal{X} to represent the variables that require optimization, $\mathcal{X} = \{\mathbf{H}_p^*, \mathbf{S}_p, \mathbf{Q}_p, \mathbf{Z}_p\}$.

Optimization and Analysis

Optimization

To solve the optimization problem in Eq.(6), we design a four-step alternating optimization algorithm.

Update \mathbf{H}_p^* Fixing the other variables, the optimization problem for \mathbf{H}_p^* can be formulated as follows:

$$\min_{\mathbf{H}_p^*} \|\mathbf{H}_p^* - \mathbf{A}_p\|_F^2, \text{ s.t. } (\mathbf{H}_p^*)^\top \mathbf{H}_p^* = \mathbf{I}_k, \quad (7)$$

where $\mathbf{A}_p = \mathbf{S}_p \mathbf{Q}_p + \mathbf{H}_{p-1}^*$. By extending the Frobenius norm into trace form, the problem of optimizing \mathbf{H}_p^* becomes:

$$\max_{\mathbf{H}_p^*} \text{Tr} \left((\mathbf{H}_p^*)^\top \mathbf{A}_p \right), \text{ s.t. } (\mathbf{H}_p^*)^\top \mathbf{H}_p^* = \mathbf{I}_k. \quad (8)$$

The optimal solution for \mathbf{H}_p^* can be obtained by computing the singular value decomposition (SVD) of \mathbf{A}_p . Specifically, if $\mathbf{A}_p = \mathbf{U}_a \Sigma_a \mathbf{V}_a^\top$, the closed-form solution (Wang et al. 2019) in Eq.(8) is as follows:

$$\mathbf{H}_p^* = \mathbf{U}_a \mathbf{V}_a^\top. \quad (9)$$

Update \mathbf{S}_p Fixing \mathbf{H}_p^* , \mathbf{Q}_p , and \mathbf{Z}_p , Eq.(6) can be expressed as follows:

$$\max_{\mathbf{S}_p} \text{Tr} \left(\mathbf{S}_p^\top \mathbf{B}_p \right), \text{ s.t. } \mathbf{S}_p^\top \mathbf{S}_p = \mathbf{I}_k, \quad (10)$$

where $\mathbf{B}_p = \mathbf{G}_p \mathbf{Z}_p + \mathbf{H}_p^* \mathbf{Q}_p^\top$. Using the same approach as described above, if $\mathbf{B}_p = \mathbf{U}_b \Sigma_b \mathbf{V}_b^\top$, the closed-form solution to Eq.(10) is obtained as follows:

$$\mathbf{S}_p = \mathbf{U}_b \mathbf{V}_b^\top. \quad (11)$$

Update \mathbf{Q}_p By eliminating irrelevant variables from the objective function, Eq.(6) can be simplified as follows:

$$\max_{\mathbf{Q}_p} \text{Tr} \left(\mathbf{Q}_p^\top \mathbf{C}_p \right), \text{ s.t. } \mathbf{Q}_p^\top \mathbf{Q}_p = \mathbf{I}_k, \quad (12)$$

where $\mathbf{C}_p = \mathbf{S}_p^\top \mathbf{H}_p^*$. Similar to Eq.(11), the closed-form solution for \mathbf{Q}_p is obtained as,

$$\mathbf{Q}_p = \mathbf{U}_c \mathbf{V}_c^\top, \quad (13)$$

where \mathbf{U}_c represents the left singular vectors of \mathbf{C}_p , and \mathbf{V}_c denotes the right singular vectors of \mathbf{C}_p .

Update \mathbf{Z}_p Fixing \mathbf{H}_p^* , \mathbf{S}_p , \mathbf{Q}_p , Eq.(6) can be simplified to,

$$\max_{\mathbf{Z}_p} \text{Tr} \left(\mathbf{Z}_p^\top \mathbf{D}_p \right), \text{ s.t. } \mathbf{Z}_p^\top \mathbf{Z}_p = \mathbf{I}_k, \quad (14)$$

Where $\mathbf{D}_p = \mathbf{G}_p^\top \mathbf{S}_p$. If $\mathbf{D}_p = \mathbf{U}_d \Sigma_d \mathbf{V}_d^\top$, Eq.(14) can be efficiently solved by SVD with computational complexity $\mathcal{O}(sk^2)$, yielding the closed-form solution,

$$\mathbf{Z}_p = \mathbf{U}_d \mathbf{V}_d^\top. \quad (15)$$

The alternating optimization process of INMKC is summarized in Algorithm 1. It is worth noting that the base kernel matrices arrive in a certain order. Upon receiving a new base kernel matrix, the most up-to-date consensus cluster structure \mathbf{H}_p^* is obtained. Performing k -means on this matrix then yields the final clustering results.

Algorithm 1: Incremental Nyström-based Multiple Kernel Clustering

Input: Datasets $\{\mathbf{X}_p\}_{p=1}^m$, landmark point number s , cluster number k , and ε_0 .
Output: \mathbf{H}_m^* .

```

1: for  $p = 1$  to  $m$  do
2:   Compute approximate base kernel matrix  $\mathbf{G}_p$  by Recursive RLS-Nyström algorithm (Musco and Musco 2017).
3:   Initialize  $\mathbf{S}_p$  and  $\mathbf{Z}_p$  by performing the truncated  $k$ -SVD on  $\mathbf{G}_p$ ,  $\mathbf{Q}_p = \mathbf{I}_k$ .
4:   if  $p = 1$  then
5:     Initialize :  $\mathbf{H}_1^* = \mathbf{S}_1$ .
6:   else
7:      $i = 1$ 
8:     while not converged do
9:       Update  $\mathbf{H}_p^*$  by solving Eq.(9).
10:      Update  $\mathbf{S}_p$  by solving Eq.(11).
11:      Update  $\mathbf{Q}_p$  by solving Eq.(13).
12:      Update  $\mathbf{Z}_p$  by solving Eq.(15).
13:       $i \leftarrow i + 1$ 
14:    end while  $(obj^{i-1} - obj^i) / obj^i \leq \varepsilon_0$ 
15:  end if
16: end for

```

Analysis and Extensions

Computational Complexity According to the optimization process described in Algorithm 1, the computation complexity of each iteration is $\mathcal{O}(k^2n) + \mathcal{O}(k^3)$. Let T denote the maximum number of iterations and m represent the number of base kernel matrices, the overall computational complexity of INMKC is $\mathcal{O}(Tms^2n) + \mathcal{O}(Tmk^2n) + \mathcal{O}(Tmk^3)$. Moreover, INMKC requires storing only the latest base kernel matrix \mathbf{G}_p and the consensus cluster structure \mathbf{H}_p^* , resulting in a storage complexity of $\mathcal{O}(sn + kn)$. When $s(s \ll n)$, both computational and storage complexity are linear with the number of samples n . Therefore, the proposed method can theoretically handle large-scale datasets.

Convergence To verify the convergence of the algorithm, we present the following proposition.

Proposition. Denote \mathcal{G} as the objective function Eq.(6). The value of \mathcal{G} is monotonically decreasing.

Proof. To prove that $\mathcal{G} = \|\mathbf{G}_p - \mathbf{S}_p \mathbf{Z}_p^\top\|_F^2 + \|\mathbf{H}_p^* - \mathbf{S}_p \mathbf{Q}_p\|_F^2 + \|\mathbf{H}_p^* - \mathbf{H}_{p-1}^*\|_F^2$ is monotonically decreasing with iterations, we need to show that each subproblem of \mathcal{G} decreases monotonically.

For the subproblem involving \mathbf{H}_p^* , at the $(i+1)$ -th iteration, given $\mathbf{S}_p^{(i)}$, $\mathbf{Q}_p^{(i)}$ and $\mathbf{Z}_p^{(i)}$, the solution $\mathbf{H}_p^{*(i+1)}$ is obtained by optimizing Eq.(8). According to Theorem 2 in (Wang et al. 2019), Eq.(8) provides a closed-form solution, ensuring that the optimization process for $\mathbf{H}_p^{*(i+1)}$ is monotonically decreasing. Similarly, for the other subproblems, it follows that the objective function \mathcal{G} is monotonically decreasing with iterations. \square

Moreover, the lower bound of the objective function Eq.(6) is zero. By the monotone convergence theorem, the algorithm is theoretically convergent. Additionally, we will verify the convergence of INMKC experimentally.

Experiment

In this section, we conduct experiments to verify the effectiveness and efficiency of INMKC. Specifically, the clustering performance, running time, kernel-fusion performance, convergence, and ablation study.

Datasets and Baselines

Dataset	Sample	View	Cluster
CUB	600	2	10
PFold	694	12	27
Fashion	10000	3	10
ALOI	10800	4	100
YTF10	38654	4	10
NMNIST	70000	2	10
YTF100	195537	4	100
Winnipeg	325834	2	7

Table 1: Description of Datasets.

Table 1 lists eight widely used public MVC datasets, including *CUB*¹, *PFold*², *Fashion* (Xiao, Rasul, and Vollgraf 2017), *ALOI*³, *YTF10*⁴, *NMNIST*⁵, *YTF100*⁶, *Winnipeg*⁷.

A summary of the nine state-of-the-art multi-view clustering algorithms is as follows:

- **LMVSC**(Kang et al. 2020) achieves large-scale spectral clustering by integrating distinct anchor graphs tailored to each view.
- **OMSC** (Chen et al. 2022) develops an efficient unified model for multi-view subspace clustering with near-linear complexity.
- **FDAGF** (Zhang et al. 2023) proposes a hybrid multi-size anchor graph fusion framework, ensuring linear complexity for large-scale datasets.
- **CAMVC**(Zhang et al. 2024b) introduces a prior cluster-guided anchor learning strategy, achieving high performance on large-scale datasets with reduced complexity.
- **SMKKM**(Liang et al. 2024b) proposes using SVD instead of eigen-decomposition (EVD) in the original SimpleMKKM algorithm, thereby reducing complexity.
- **OLICAG**(Zhang et al. 2024a) presents a strategy for learning cross-view anchor graphs, enhancing multi-view fuzzy clustering through latent information fusion.

¹<http://www.vision.caltech.edu/visipedia/CUB-200.html>

²mkl.csdl.tamu.edu/dataset/protein-fold-prediction

³<https://www.kaggle.com/alvations/alois-dataset>

⁴<https://www.cs.tau.ac.il/~wolf/ytfaces/>

⁵<http://yann.lecun.com/exdb/mnist/>

⁶<https://www.cs.tau.ac.il/~wolf/ytfaces/>

⁷<https://archive.ics.uci.edu/ml/datasets/Crop+mapping+using+fused+optical-radar+data+set>

- **IMSC** (Zhou et al. 2019) integrates incremental learning into multi-view spectral clustering, updating a consensus model by sequentially fusing views.
- **SCGL** (Yin et al. 2021) combines sparse graph learning with connected graph learning to propose an efficient incremental multi-view spectral clustering method.
- **CMVC**(Wan et al. 2022) requires maintaining only a consensus partition matrix and updates knowledge with new view data for clustering.

Experiment Setup

The implementation codes of the aforementioned methods are publicly available in their respective papers, and we run them without any modifications. For methods with hyperparameters, we tune them using grid search as recommended in their papers to achieve the best outcomes. In our experiment, the dimension of the Nyström approximation matrix is set to $s = 300$. Considering that all algorithms eventually require performing k -means to obtain final clustering assignments, we repeat k -means 50 times to reduce the randomness and report the maximum values in Table 2. We employ three evaluation metrics, including Accuracy (ACC), Normalized Mutual Information (NMI), and Purity, to verify the clustering performance. All experiments are conducted on a computer equipped with an Intel Core i9-9900K CPU and 48GB RAM.

Clustering Performance

Table 2 presents the clustering results on eight benchmark datasets. The best value is marked in bold, and '-' indicates that the algorithm failed to run due to insufficient memory. From the table, we observe that:

- Our proposed method outperforms state-of-the-art competitors on PFold, ALOI, YTFace10, NMNIST, YTF100 and Winnipeg. INMKC also achieves significant improvements on other datasets. Especially, INMKC exceeds the performance of the second-best algorithm by 1.83%, 1.73%, 3.21%, 2.71%, 6.15%, 5.82%, 1.22% and 0.24% in ACC across all benchmark datasets.
- As a strong baseline for large-scale MKC algorithm, SMKKM(Liang et al. 2024b) achieves high performance. However, on all benchmark datasets, INMKC consistently outperforms SMKKM by 20.16%, 10.95%, 6.14%, 24.42%, 20.57% and 20.44% in terms of ACC. This significant performance gap demonstrates the effectiveness of our approximation method based on leverage score sampling.
- As a powerful baseline for MVC algorithm based on incremental learning, CMVC(Wan et al. 2022) also demonstrates high clustering performance. However, CMVC requires the selection of a hyperparameter, which can limit its practical application due to the absence of ground truth in clustering tasks. In contrast, INMKC is parameter-free and outperforms CMVC by 3.16%, 2.74%, 4.25%, 2.71%, 11.04%, 5.82%, 1.34% and 4.51% in terms of ACC across all benchmark datasets.

Dataset	Metric	LMVSC	OMSC	FDAGF	CAMVC	SMKKM	OLICAG	IMSC	SCGL	CMVC	INMKC
CUB	ACC	74.67	65.17	74.00	79.50	61.17	61.50	68.17	78.00	78.17	81.33
	NMI	71.06	73.50	74.02	74.72	58.26	62.21	63.84	78.32	78.05	76.50
	Purity	75.17	67.67	81.33	79.50	64.83	61.50	68.83	78.50	79.50	81.33
PFold	ACC	22.48	28.67	29.68	36.02	26.80	19.31	20.61	22.33	35.01	37.75
	NMI	26.52	34.08	38.28	45.15	35.00	25.40	33.14	28.19	40.04	46.53
	Purity	42.51	31.41	44.09	42.36	34.15	23.63	29.68	27.81	42.07	45.10
Fashion	ACC	54.62	64.44	68.95	73.88	70.95	63.84	63.62	68.07	72.84	77.09
	NMI	51.52	73.75	69.78	74.14	63.94	61.39	68.57	75.72	71.11	72.04
	Purity	60.41	67.93	76.38	76.00	70.95	64.71	71.92	69.27	72.87	77.09
ALOI	ACC	57.53	34.20	54.42	62.32	49.27	66.15	49.94	61.69	70.98	73.69
	NMI	75.42	68.44	75.00	75.29	65.72	78.17	70.21	76.89	82.61	83.67
	Purity	70.53	35.32	73.70	63.29	53.76	67.27	56.60	63.08	74.22	75.27
YTF10	ACC	75.42	74.04	73.04	82.86	68.44	-	-	-	77.97	89.01
	NMI	77.83	76.91	79.24	84.62	74.48	-	-	-	80.47	86.47
	Purity	80.91	76.16	81.27	86.67	76.43	-	-	-	81.30	89.01
NMNIST	ACC	39.11	58.49	47.94	42.39	45.29	-	-	-	59.91	65.73
	NMI	33.53	51.60	47.25	36.52	34.75	-	-	-	51.62	54.62
	Purity	43.22	59.73	62.81	47.04	46.02	-	-	-	63.38	65.74
YTF100	ACC	61.07	63.85	62.00	68.23	-	-	-	-	68.11	69.45
	NMI	78.52	81.55	80.26	82.21	-	-	-	-	83.48	84.04
	Purity	71.02	68.02	75.08	73.67	-	-	-	-	74.95	76.70
Winnipeg	ACC	56.85	52.16	60.07	54.33	-	-	-	-	55.80	60.31
	NMI	41.60	45.39	53.06	46.19	-	-	-	-	42.28	50.51
	Purity	57.11	63.51	63.66	68.86	-	-	-	-	68.72	76.49

Table 2: Empirical evaluation and comparison of INMKC with nine baseline methods on eight benchmark datasets in terms of clustering accuracy (ACC), normalized mutual information (NMI), and Purity.

Running Time

We evaluate the time efficiency of the INMKC algorithm, as illustrated in Figure 3. The reported running time encompasses the initialization phase for all algorithms. The running time of the INMKC algorithm is measured from the incorporation of the first base kernel matrix until fusion with the last one, consistent with the linear time complexity. Experimental results indicate that the INMKC algorithm has a shorter running time compared to most benchmark methods, confirming its computational efficiency.

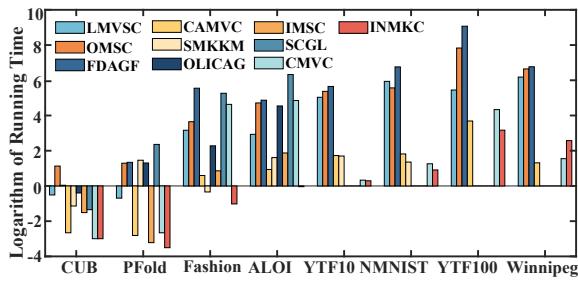


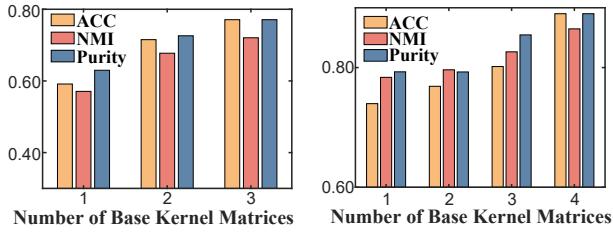
Figure 3: Comparison of INMKC with nine benchmark methods in terms of relative logarithmic running time on eight benchmark datasets. A missing bar indicates that the method encountered an 'out-of-memory' error.

Kernel-Fusion Performance

Given that INMKC can handle scenarios where the kernel matrix grows over time, we conduct experiments to examine whether INMKC enhances the quality of information fusion as the number of kernel matrices increases incrementally. In the experiment, we record clustering results with each added kernel matrix. Due to space limitations, we present results from only two datasets. As shown in Figure 4, clustering performance improves with the increasing number of kernel matrices, demonstrating that INMKC effectively fuses information from previous kernel matrices.

Convergence

In the previous section, we have proven the convergence of the algorithm theoretically. Subsequent experiments further validate that the algorithm eventually converges to a local optimum. We plot the changes in the objective value of INMKC over iterations for the Fashion and YTFace10 datasets, similar results for other datasets are omitted due to space limitations. As shown in Figure 5, the objective value decreases monotonically, and the algorithm converges within 10 iterations.



(a) Fashion

(b) YTFace10

Figure 4: The kernel-fusion performance of INMKC with kernel matrices collected in order on two datasets. Similar results for other datasets are omitted due to space constraints.

Method		INMKC		CMVC	
CD	Dataset	LS	US	LS	US
Uniform	CUB	81.33	78.83	81.00	78.33
	Fashion	77.09	77.08	69.91	70.65
	ALOI	73.69	73.17	71.30	69.89
	NMNIST	65.73	57.15	64.80	63.41
Unbalanced	PFold	37.75	36.89	36.02	32.56
	YTF10	89.01	77.58	80.79	74.67
	YTF100	69.45	68.72	69.00	67.75
	Winnipeg	60.31	54.28	59.73	57.46

Table 3: The ablation study of the sampling method conducted by INMKC and CMVC on eight benchmark datasets in terms of ACC. The best results are marked in bold. ‘CD’ represents cluster distribution, ‘LS’ represents leverage score sampling, and ‘US’ represents uniform sampling.

Ablation Study

INMKC introduces two significant innovations in large-scale MKC methods: leverage score sampling and incremental learning. To validate these contributions, we conduct two ablation experiments.

In the first experiment, we compare leverage score sampling with uniform sampling for both INMKC and CMVC (Wan et al. 2022) algorithms. Results in Table 3 show that leverage score sampling improves clustering performance, especially on datasets with unbalanced cluster distributions, such as Winnipeg, where the smallest cluster comprises only 0.35%. The use of leverage score sampling significantly enhances clustering performance, suggesting it can improve the efficiency and effectiveness of large-scale MVC algorithms.

In the second experiment, to verify the effectiveness of incremental learning, we provide the objective function without incremental learning, referred to as NMKC.

$$\begin{aligned} \min_{\mathbf{H}^*, \mathbf{S}_p, \mathbf{Z}_p, \mathbf{Q}_p} & \sum_{p=1}^m \left(\|\mathbf{G}_p - \mathbf{S}_p \mathbf{Z}_p^\top\|_F^2 + \|\mathbf{H}^* - \mathbf{S}_p \mathbf{Q}_p\|_F^2 \right), \\ \text{s.t. } & \mathbf{S}_p^\top \mathbf{S}_p = \mathbf{I}_k, \quad \mathbf{Z}_p^\top \mathbf{Z}_p = \mathbf{I}_k, \quad (\mathbf{H}^*)^\top \mathbf{H}^* = \mathbf{I}_k, \quad \mathbf{Q}_p^\top \mathbf{Q}_p = \mathbf{I}_k, \end{aligned} \quad (16)$$

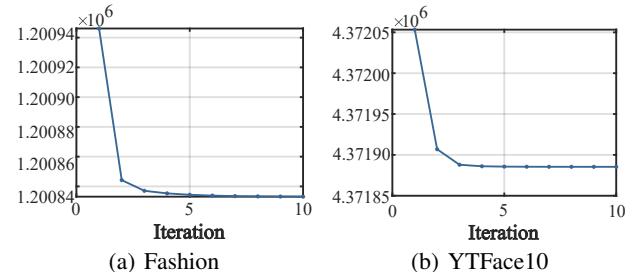


Figure 5: The objective value of INMKC varies with increasing iterations. Similar results for other datasets are omitted due to space limitations.

where \mathbf{H}^* represents the consensus clustering structure matrix. The results in Table 4 indicate that incremental learning not only enhances the fusion of base kernel information, improving clustering performance, but also significantly increases computational efficiency. Although INMKC has slightly lower ACC than NMKC on the Winnipeg datasets, its overall clustering performance is superior.

Method	INMKC		NMKC	
	Dataset	ACC	time	ACC
CUB	81.33	0.05	53.67	0.29
PFold	37.75	0.03	35.01	0.91
Fashion	77.09	0.36	49.95	2.33
ALOI	73.69	0.96	59.81	6.87
YTF10	89.01	1.34	87.62	11.97
NMNIST	65.73	2.48	51.37	10.66
YTF100	69.45	23.73	67.75	104.60
Winnipeg	60.31	13.11	63.34	37.76

Table 4: The ablation study of INMKC and NMKC on eight benchmark datasets in terms of ACC and time. The best results are marked in bold. Due to space limitations, only the results for ACC are shown.

Conclusion

In this paper, we address the limitations of existing MKC algorithms and propose a parameter-free method, termed Incremental Nyström-based Multiple Kernel Clustering(INMKC). INMKC integrates matrix decomposition and incremental learning into a unified framework. When a new base kernel matrix emerges, INMKC first approximates it using the Nyström method with leverage score sampling. It then performs matrix factorization and updates the consensus cluster structure. These processes enable the algorithm to handle both large-scale and domain-incremental dynamic datasets efficiently. Our extensive experiments validate the effectiveness and efficiency of INMKC, demonstrating its superior performance compared to existing MKC algorithms. Furthermore, the design of INMKC significantly reduces computational overhead, making it a practical and scalable solution for real-world applications.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant No. 62325604, 62276271, 62306324, 62376279, U24A20333), the Science and Technology Innovation Program of Hunan Province (Grant No. 2024RC3128) and National University of Defense Technology Research Foundation (No. ZK24-30).

References

- Alaoui, A.; and Mahoney, M. W. 2015. Fast randomized kernel ridge regression with statistical guarantees. *Advances in neural information processing systems*, 28: 775–783.
- Cai, H.; Tan, Y.; Huang, S.; and Lv, J. 2023. Lifelong Multi-view Spectral Clustering. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, 3488–3496.
- Chao, G.; Sun, S.; and Bi, J. 2021. A Survey on Multiview Clustering. *IEEE Transactions on Artificial Intelligence*, 2(2): 146–168.
- Chen, M.-S.; Wang, C.-D.; Huang, D.; Lai, J.-H.; and Yu, P. S. 2022. Efficient orthogonal multi-view subspace clustering. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, 127–135.
- Chen, X.; and Cai, D. 2011. Large Scale Spectral Clustering with Landmark-Based Representation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 25(1): 313–318.
- Du, L.; Zhou, P.; Shi, L.; Wang, H.; Fan, M.; Wang, W.; and Shen, Y.-D. 2015. Robust multiple kernel k-means using l21-norm. In *Twenty-fourth international joint conference on artificial intelligence*, 3476–3482.
- Han, D.; and Kim, J. 2015. Unsupervised simultaneous orthogonal basis clustering feature selection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5016–5023.
- Hofmann, T.; Schölkopf, B.; and Smola, A. J. 2008. Kernel methods in machine learning. *The Annals of Statistics*, 36(3): 1171 – 1220.
- Hu, M.; and Chen, S. 2019. One-pass incomplete multi-view clustering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, 3838–3845.
- Huang, H.-C.; Chuang, Y.-Y.; and Chen, C.-S. 2011. Multiple kernel fuzzy clustering. *IEEE Transactions on Fuzzy Systems*, 20(1): 120–134.
- Kang, Z.; Zhou, W.; Zhao, Z.; Shao, J.; Han, M.; and Xu, Z. 2020. Large-scale multi-view subspace clustering in linear time. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 4412–4419.
- Liang, K.; Meng, L.; Liu, Y.; Liu, M.; Wei, W.; Liu, S.; Tu, W.; Wang, S.; Zhou, S.; and Liu, X. 2024a. Simple Yet Effective: Structure Guided Pre-trained Transformer for Multi-modal Knowledge Graph Reasoning. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 1554–1563.
- Liang, W.; Liu, X.; Liu, Y.; Ma, C.; Zhao, Y.; Liu, Z.; and Zhu, E. 2023. Consistency of multiple kernel clustering. In *International Conference on Machine Learning*, 20650–20676. PMLR.
- Liang, W.; Tang, C.; Liu, X.; Liu, Y.; Liu, J.; Zhu, E.; and He, K. 2024b. On the Consistency and Large-Scale Extension of Multiple Kernel Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–13.
- Liang, W.; Zhu, E.; Yu, S.; Xu, H.; Zhu, X.; and Liu, X. 2024c. Scalable Multiple Kernel Clustering: Learning Clustering Structure from Expectation. In *Forty-first International Conference on Machine Learning*.
- Liu, X. 2023. Hyperparameter-free localized simple multiple kernel K-means with global optimum. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(7): 8566–8576.
- Lu, Y.; Xin, H.; Wang, R.; Nie, F.; and Li, X. 2022. Scalable multiple kernel k-means clustering. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 4279–4283.
- Musco, C.; and Musco, C. 2017. Recursive sampling for the nystrom method. *Advances in neural information processing systems*, 30: 3836–3848.
- Qu, Q.; Wan, X.; Liang, W.; Liu, J.; Feng, Y.; Xu, H.; Liu, X.; and Zhu, E. 2024. A Lightweight Anchor-Based Incremental Framework for Multi-view Clustering. In *Proceedings of the 32nd ACM International Conference on Multimedia, MM ’24*, 8652–8661. New York, NY, USA: Association for Computing Machinery. ISBN 9798400706868.
- van der Maaten, L.; and Hinton, G. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(86): 2579–2605.
- Vergara, A.; Fonollosa, J.; Mahiques, J.; Trincavelli, M.; Rulkov, N.; and Huerta, R. 2013. On the performance of gas sensor arrays in open sampling systems using Inhibitory Support Vector Machines. *Sensors and Actuators B: Chemical*, 185: 462–477.
- Wan, X.; Liu, J.; Gan, X.; Liu, X.; Wang, S.; Wen, Y.; Wan, T.; and Zhu, E. 2024a. One-Step Multi-View Clustering With Diverse Representation. *IEEE Transactions on Neural Networks and Learning Systems*, 1–13.
- Wan, X.; Liu, J.; Liang, W.; Liu, X.; Wen, Y.; and Zhu, E. 2022. Continual multi-view clustering. In *Proceedings of the 30th ACM International Conference on Multimedia*, 3676–3684.
- Wan, X.; Xiao, B.; Liu, X.; Liu, J.; Liang, W.; and Zhu, E. 2024b. Fast Continual Multi-View Clustering With Incomplete Views. *IEEE Transactions on Image Processing*, 33: 2995–3008.
- Wang, L.; Zhang, X.; Su, H.; and Zhu, J. 2023. A Comprehensive Survey of Continual Learning: Theory, Method and Application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46: 5362–5383.
- Wang, S.; Liu, X.; Zhu, E.; Tang, C.; Liu, J.; Hu, J.; Xia, J.; and Yin, J. 2019. Multi-view Clustering via Late Fusion Alignment Maximization. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, 3778–3784.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. *ArXiv*, abs/1708.07747.

Yang, X.; Jiaqi, J.; Wang, S.; Liang, K.; Liu, Y.; Wen, Y.; Liu, S.; Zhou, S.; Liu, X.; and Zhu, E. 2023. Dealmvc: Dual contrastive calibration for multi-view clustering. In *Proceedings of the 31st ACM International Conference on Multimedia*, 337–346.

Yang, Y.; and Wang, H. 2018. Multi-view clustering: A survey. *Big data mining and analytics*, 1(2): 83–107.

Yin, H.; Hu, W.; Zhang, Z.; Lou, J.; and Miao, M. 2021. Incremental multi-view spectral clustering with sparse and connected graph learning. *Neural Networks*, 144: 260–270.

Zhang, C.; Chen, L.; Shi, Z.; and Ding, W. 2024a. Latent information-guided one-step multi-view fuzzy clustering based on cross-view anchor graph. *Inf. Fusion*, 102(C).

Zhang, C.; Jia, X.; Li, Z.; Chen, C.; and Li, H. 2024b. Learning Cluster-Wise Anchors for Multi-View Clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 16696–16704.

Zhang, P.; Wang, S.; Li, L.; Zhang, C.; Liu, X.; Zhu, E.; Liu, Z.; Zhou, L.; and Luo, L. 2023. Let the data choose: Flexible and diverse anchor graph fusion for scalable multi-view clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 11262–11269.

Zhao, B.; Kwok, J. T.; and Zhang, C. 2009. Multiple kernel clustering. In *Proceedings of the 2009 SIAM international conference on data mining*, 638–649. SIAM.

Zhou, P.; Shen, Y.-D.; Du, L.; Ye, F.; and Li, X. 2019. Incremental multi-view spectral clustering. *Knowledge-Based Systems*, 174: 73–86.