

Article

Sparse Multiple Kernel Concept Factorization with Adaptive Orthogonal Factors

Qiang Shu ¹, Yan Chen ², Yunhui Liang ³ and Liang Du ^{3,*} ¹ Shanxi Science and Technology Achievements Transfer and Transformation Promotion and Data Monitoring Center, Taiyuan 030024, China² College of Computer Science and Technology, Taiyuan University of Technology, Taiyuan 030600, China³ Institute of Big Data Science and Industry, Shanxi University, Taiyuan 030006, China

* Correspondence: duliang@sxu.edu.cn

Abstract

Concept factorization provides an interpretable route to clustering by expressing latent concepts as combinations of observed samples. Its development in multiple kernel settings, however, remains limited when one simultaneously requires kernel validity, low storage, and mathematically transparent optimization. This paper proposes a sparse multiple kernel concept factorization framework with adaptive orthogonal factors. The framework first constructs sparse positive semi-definite kernel matrices through localized neighborhood regression and then learns a shared nonnegative consensus representation together with kernel-specific orthogonal factors and adaptive kernel weights. The resulting block coordinate procedure contains two closed-form subproblems and one standard multiplicative update. The positive semi-definiteness of the sparse kernels is proved, the exact solutions of the orthogonality-constrained and weight-update blocks are derived, monotonic descent, lower boundedness, and the limiting behavior of the generated objective sequence are analyzed. The computational and storage complexity of the framework are also analyzed. Experiments on nine benchmark datasets, whose sample sizes range from 10^3 to 2.8×10^5 , show that the proposed method remains competitive in clustering quality while retaining attractive runtime and memory behavior. The paper therefore offers a mathematically explicit and empirically verifiable treatment of sparse multiple kernel concept factorization.

Keywords: multiple kernel clustering; concept factorization; sparse kernel construction; orthogonal factors; adaptive weighting; block coordinate optimization

MSC: 68T10



Academic Editors: Gang Wang, Jing Liu and Jingling Ma

Received: 21 May 2026

Revised: 14 June 2026

Accepted: 25 June 2026

Published: 2 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Kernel methods remain an important family of techniques for unsupervised learning with nonlinear data. By implicitly mapping samples into high-dimensional feature spaces, a kernel representation can reveal cluster structures that may be difficult to recover in the original input domain. In practice, however, the performance of a single kernel often depends strongly on the selected kernel type and its associated parameters. Multiple kernel clustering (MKC) addresses this limitation by combining several candidate kernels and has become a standard paradigm for robust clustering under heterogeneous similarity descriptions [1–3].

Despite its empirical success, MKC still faces an evident tension between representation quality and computational feasibility. Early-fusion formulations usually construct a

consensus kernel or a consensus partition directly from dense kernel matrices [3,4]. These approaches can be accurate, but their memory requirements and repeated spectral computations become increasingly restrictive as the sample size grows. Late-fusion methods alleviate some of this burden by first learning kernel-specific partitions and then combining them at the partition level [5–7]. Yet the decoupling between kernel-specific learning and consensus formation may weaken useful cross-kernel interactions. Sampling-based or anchor-based variants improve scalability further, but they may also introduce approximation artifacts or sensitivity to the chosen subset size [8–10].

Concept factorization (CF) offers a different viewpoint. Instead of clustering directly from kernel eigenvectors, CF seeks latent concepts that are themselves represented as combinations of observed samples [11]. This perspective is attractive because it yields interpretable low-dimensional representations and naturally connects clustering with matrix factorization. Orthogonality-enhanced variants further improve the identifiability and discriminative behavior of the latent factors [12]. Recent multi-view CF and matrix-factorization models further show that weighting, graph regularization, deep factorization, and diversity constraints can all improve clustering quality when they are incorporated into one structured model [13–18]. Nevertheless, a mathematically simple and computationally efficient multiple-kernel CF framework is still underdeveloped.

This paper studies such a framework. Different from recent scalable MKC methods that mainly reduce complexity through sampling, anchors, expectation-based approximation, or low-rank acceleration, the proposed framework pursues scalability through sparse positive semi-definite kernel operators. Our starting point is a sparse kernel construction based on localized neighborhood regression. The sparse kernels are not used as a preprocessing shortcut alone; instead, they are directly coupled with a multiple-kernel orthogonal concept factorization model, where a shared nonnegative consensus factor, kernel-specific orthogonal factors, and adaptive kernel weights are jointly optimized. This design distinguishes the proposed method from existing scalable MKC approaches that typically construct kernel-specific partitions or reduced kernel representations before consensus learning. The purpose of this construction is twofold: it reduces storage by retaining only the strongest local relations, and it preserves a valid positive semi-definite kernel structure. Based on these sparse kernels, we formulate a weighted multiple-kernel orthogonal concept factorization model. The model learns a shared nonnegative consensus representation $\mathbf{U}$ together with kernel-specific orthogonal factors $\{\mathbf{H}_r\}_{r=1}^m$ and adaptive kernel weights $\{\mu_r\}_{r=1}^m$. The shared representation plays the role of a consensus latent clustering structure, while the orthogonal factors preserve kernel-dependent information in a compact and numerically stable form.

The present study therefore emphasizes a mathematically structured route to multiple-kernel clustering. The sparse kernel acts as a valid and economical operator, the optimization variables are chosen so that every block remains tractable, and the resulting algorithm is supported by transparent analysis of kernel validity, block optimality, monotonic descent, limiting behavior, and complexity. The paper should thus be understood primarily as a model-and-optimization study for multiple-kernel concept factorization rather than as a purely engineering-oriented scalability report.

This perspective is important because the value of the proposed framework lies in the coherence of its components. The sparse-kernel construction, the weighted orthogonal factorization model, and the blockwise optimization procedure are designed to fit together within one mathematically explicit pipeline. The corresponding experiments are then used to verify that this pipeline is not only analytically sound but also practically effective on benchmark data.

The main contributions of this paper are summarized as follows.

- We propose a sparse multiple-kernel concept factorization model that extends orthogonal concept factorization to the MKC setting through a shared nonnegative consensus representation and kernel-specific orthogonal factors.
- We introduce a localized sparse kernel construction and show that the resulting sparse kernel remains positive semi-definite.
- We derive a block coordinate optimization algorithm in which the orthogonal-factor update and the kernel-weight update admit closed-form solutions, while the consensus representation is updated through a standard multiplicative rule.
- We provide a concise theoretical analysis covering kernel validity, block optimality, monotonic descent, lower boundedness, objective-value convergence, and computational complexity.
- We report extensive experiments on nine benchmark datasets to show that the proposed framework is competitive in clustering quality and remains efficient in both runtime and memory consumption.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 introduces the sparse kernel construction and the proposed multiple-kernel concept factorization model. Section 4 presents the optimization algorithm. Section 5 provides the theoretical analysis. Section 6 reports the experimental results. Section 7 concludes the paper.

2. Related Work

2.1. Multiple Kernel Clustering

MKC integrates several candidate kernels to recover a consensus clustering structure. Classical formulations directly learn a combined kernel or a combined partition from dense kernel matrices [1,2,4]. Although such methods are often effective, they usually require dense kernel storage and repeated eigenvalue decompositions, which limits their applicability in large-scale settings. Discrete or parameter-free variants reduce some of the spectral burden, but dense kernels are still typically required during optimization [3,19].

To improve scalability, researchers have proposed late-fusion, sampling-based, and expectation-driven routes. Late-fusion methods first obtain kernel-specific partitions or embeddings and then perform consensus learning in a second stage [5–7,20,21]. This strategy lowers the cost of the consensus step, but the decoupling between base learning and fusion may weaken structural interactions across kernels. Sampling-based and anchor-based methods, such as CMKC, SMKC, and SVDSMKKM, further reduce the cost of dense kernel processing by working with subsets or reduced kernel matrices [8–10]. Recent MKC and MKL studies have also investigated complementary mechanisms such as discrete partition-balance regularization and topology-aware ensemble learning [22,23]. More recent MKC work has also explored differentiable or learned kernel connections beyond the classical fixed-kernel setting [24]. However, the use of subsets, anchors, or auxiliary structural constraints may affect structural fidelity when the reduced or regularized representation is insufficiently expressive.

2.2. Concept Factorization and Its Extensions

Concept factorization represents latent concepts as combinations of observed samples and provides an interpretable factorization view of clustering [11]. This idea has motivated a variety of developments, including orthogonality-enhanced concept factorization [12], graph-regularized variants, and multi-view extensions [13,15,18]. Such methods show that factorization-based clustering can benefit from additional structural constraints, especially when multiple views or multiple sources of similarity are available.

It is worth noting that some recent deep factorization and multi-view factorization models are not directly comparable with the present method because they usually operate on multi-view feature matrices or learned representations rather than on multiple pre-computed kernel operators. Recent multi-view clustering models with mixture-of-experts, view balance, and regional sparsity also belong to this broader feature-level multi-view learning setting [25]. They are reviewed here to position the factorization and multi-view learning perspectives, not as direct scalable MKC baselines. Accordingly, the experimental comparison focuses on representative MKC methods that share the same multiple-kernel input setting. Nevertheless, existing CF-based methods do not fully address the particular difficulties of multiple-kernel clustering. In the multi-kernel setting, one needs to combine several kernel operators while keeping the model computationally manageable and mathematically interpretable. This motivates the present work: we use sparse valid kernels as the input operators and construct a multiple-kernel orthogonal concept factorization framework with adaptive weighting.

3. Sparse Kernel Construction and Model Formulation

3.1. Problem Setting

Let $\mathbf{X} = [x_1, x_2, \dots, x_n] \in \mathbb{R}^{d \times n}$ denote a dataset of n samples. Suppose that m candidate kernels are available. The r th kernel produces a Gram matrix $\mathbf{K}_r^o \in \mathbb{R}^{n \times n}$, where $\mathbf{K}_r^o(i, j)$ measures the similarity between x_i and x_j under the r th kernel.

Our goal is to construct a mathematically structured factorization model on top of a sparse collection of kernels. We denote by c the number of clusters and by k the neighborhood size used in the sparse-kernel construction. Throughout the paper, k is the sparsification parameter and c is the cluster number.

3.2. Notation Convention

Scalars are written in italic form, for example, m , n , c , k , and t . Matrices are written in bold uppercase form, for example, $\mathbf{X}$, $\mathbf{K}_r$, $\mathbf{U}$, and $\mathbf{H}_r$. Vector-valued quantities such as the kernel-weight vector $\boldsymbol{\mu} = (\mu_1, \dots, \mu_m)^\top$ are written in bold lowercase Greek form. Calligraphic symbols denote sets, such as the neighborhood set $\mathcal{N}_i$. For ease of reference, Table 1 summarizes the core notation used in the paper.

Table 1. Core notation used in the paper.

Symbol	Meaning
$\mathbf{X} = [x_1, \dots, x_n] \in \mathbb{R}^{d \times n}$	data matrix
$\mathbf{K}_r^o, \mathbf{K}_r \in \mathbb{R}^{n \times n}$	original and sparse kernels for the r th source
$\mathbf{S}_r, \mathbf{A}_r, \mathbf{D}_r$	local coefficient, symmetrized affinity, and degree matrices
$\mathbf{U} \in \mathbb{R}^{n \times c}$	shared nonnegative consensus factor
$\mathbf{H}_r \in \mathbb{R}^{n \times c}$	orthogonal factor for the r th kernel
$\boldsymbol{\mu} = (\mu_1, \dots, \mu_m)^\top$	adaptive kernel-weight vector
c and k	cluster number and neighborhood size

3.3. Sparse Kernel Construction

Dense kernel matrices are expensive to store and manipulate for large n . To reduce this burden while retaining meaningful local structure, we construct a sparse kernel from each dense kernel matrix. Consider an original kernel matrix $\mathbf{K}^o$. For each sample x_i , let $\mathcal{N}_i = \{x_{j_1}, x_{j_2}, \dots, x_{j_k}\}$ denote the set of its k strongest neighbors according to the entries of $\mathbf{K}^o$.

Motivated by the Nadaraya–Watson estimator [26,27], we define the local coefficient matrix $\mathbf{S} \in \mathbb{R}^{n \times n}$ by

$$s_{ij} = \begin{cases} \frac{\kappa_{ij}}{\sum_{x_{j'} \in \mathcal{N}_i} \kappa_{ij'}}, & x_j \in \mathcal{N}_i, \\ 0, & \text{otherwise,} \end{cases} \quad (1)$$

where $\kappa_{ij} = \mathbf{K}^o(i, j)$. Clearly, $\mathbf{S} \geq 0$ entrywise and $\mathbf{S}\mathbf{1} = \mathbf{1}$. Since $\mathbf{S}$ is generally asymmetric, we symmetrize it by

$$\mathbf{A} = \frac{\mathbf{S} + \mathbf{S}^\top}{2}. \quad (2)$$

Let $\mathbf{D}$ be the degree matrix of $\mathbf{A}$, that is, $\mathbf{D}(i, i) = \sum_{j=1}^n a_{ij}$. We then define the sparse kernel by

$$\mathbf{K} = \frac{1}{2}(\mathbf{I} + \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}). \quad (3)$$

Lemma 1. *The matrix $\mathbf{K}$ defined in (3) is symmetric positive semi-definite.*

Proof. The matrix $\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ is symmetric. Moreover, the normalized Laplacian

$$\mathbf{L}_{\text{sym}} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$$

is positive semi-definite. Hence, all eigenvalues of $\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ belong to $[-1, 1]$. It follows that every eigenvalue of

$$\frac{1}{2}(\mathbf{I} + \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2})$$

lies in $[0, 1]$. Therefore, $\mathbf{K}$ is positive semi-definite. $\square$

Remark 1. *The sparse kernel in (3) retains only the strongest local interactions while avoiding full dense storage. Its construction still requires neighbor identification, but the resulting representation only stores $O(nk)$ nonzero coefficients rather than $O(n^2)$ entries.*

The use of the k strongest neighbors intentionally emphasizes local relations. This may discard some weak long-range connections, especially when the global cluster geometry is highly non-convex. However, the sparse kernel is constructed separately for each candidate kernel, and the subsequent multiple-kernel factorization aggregates complementary local structures across kernels. Thus, the construction should be understood as a locality-preserving and memory-saving approximation rather than a universally optimal graph construction. The influence of k is empirically examined in Section 6.

Applying this construction to each original kernel $\mathbf{K}_r^o$ produces a sparse positive semi-definite kernel $\mathbf{K}_r$, $r = 1, \dots, m$.

3.4. Multiple-Kernel Orthogonal Concept Factorization

Concept factorization reconstructs data through latent concepts expressed as combinations of observed samples [11]. In the kernel setting, an orthogonality-enhanced formulation can be written as

$$\min_{\mathbf{U}, \mathbf{H}} \text{tr}(\mathbf{K}) - 2\text{tr}(\mathbf{U}^\top \mathbf{K} \mathbf{H}) + \text{tr}(\mathbf{U}^\top \mathbf{K} \mathbf{U}) \quad \text{s.t.} \quad \mathbf{U} \geq 0, \mathbf{H}^\top \mathbf{H} = \mathbf{I}_c, \quad (4)$$

where $\mathbf{U} \in \mathbb{R}^{n \times c}$ is a nonnegative coefficient matrix and $\mathbf{H} \in \mathbb{R}^{n \times c}$ is an orthogonality-constrained factor.

To extend (4) to the multiple-kernel setting, we propose the following objective:

$$\min_{\mathbf{U}, \{\mathbf{H}_r\}, \boldsymbol{\mu}} \sum_{r=1}^m \frac{1}{\mu_r} \left(\text{tr}(\mathbf{K}_r) - 2 \text{tr}(\mathbf{U}^\top \mathbf{K}_r \mathbf{H}_r) + \text{tr}(\mathbf{U}^\top \mathbf{K}_r \mathbf{U}) \right) \quad (5)$$

subject to

$$\mathbf{U} \geq 0, \quad \mathbf{H}_r^\top \mathbf{H}_r = \mathbf{I}_c, \quad r = 1, \dots, m, \quad \sum_{r=1}^m \mu_r = 1, \quad \mu_r \geq 0. \quad (6)$$

In this model, $\mathbf{U} \in \mathbb{R}^{n \times c}$ is a shared nonnegative consensus representation, $\mathbf{H}_r \in \mathbb{R}^{n \times c}$ is the orthogonal factor associated with the r th kernel, and μ_r is the adaptive weight assigned to that kernel. The inverse weighting $1/\mu_r$ allows more informative kernels to contribute more strongly to the objective through smaller reconstruction loss.

The model in (5) and (6) has a clear interpretation. The shared factor $\mathbf{U}$ captures the consensus latent structure across kernels, whereas each orthogonal factor $\mathbf{H}_r$ absorbs kernel-specific information in a compact and well-conditioned way. The orthogonality constraints remove part of the ambiguity that would otherwise appear in fully nonnegative bilinear factorizations.

4. Optimization Algorithm

The variables in (5) and (6) are optimized by block coordinate descent. Each iteration consists of three updates.

4.1. Update of $\mathbf{H}_r$

With $\mathbf{U}$ and $\boldsymbol{\mu}$ fixed, the subproblem for each $\mathbf{H}_r$ is

$$\max_{\mathbf{H}_r} \text{tr}(\mathbf{H}_r^\top \mathbf{E}_r) \quad \text{s.t.} \quad \mathbf{H}_r^\top \mathbf{H}_r = \mathbf{I}_c, \quad (7)$$

where

$$\mathbf{E}_r = \mathbf{K}_r \mathbf{U}. \quad (8)$$

Let the thin singular value decomposition of $\mathbf{E}_r$ be

$$\mathbf{E}_r = \mathbf{P}_r \boldsymbol{\Sigma}_r \mathbf{Q}_r^\top. \quad (9)$$

Then the optimal solution of (7) is

$$\mathbf{H}_r = \mathbf{P}_r \mathbf{Q}_r^\top. \quad (10)$$

This is the classical orthogonal Procrustes solution.

4.2. Update of $\boldsymbol{\mu}$

With $\mathbf{U}$ and $\{\mathbf{H}_r\}$ fixed, define

$$\xi_r = \text{tr}(\mathbf{K}_r) - 2 \text{tr}(\mathbf{U}^\top \mathbf{K}_r \mathbf{H}_r) + \text{tr}(\mathbf{U}^\top \mathbf{K}_r \mathbf{U}). \quad (11)$$

The weight-update subproblem becomes

$$\min_{\boldsymbol{\mu}} \sum_{r=1}^m \frac{\xi_r}{\mu_r} \quad \text{s.t.} \quad \sum_{r=1}^m \mu_r = 1, \quad \mu_r \geq 0. \quad (12)$$

Proposition 1. The optimal solution of (12) is

$$\mu_r = \frac{\sqrt{\xi_r}}{\sum_{q=1}^m \sqrt{\xi_q}}, \quad r = 1, \dots, m. \quad (13)$$

Proof. The Lagrangian of (12) is

$$\mathcal{L}(\mu, \lambda) = \sum_{r=1}^m \frac{\xi_r}{\mu_r} + \lambda \left(\sum_{r=1}^m \mu_r - 1 \right).$$

Stationarity with respect to μ_r gives

$$-\frac{\xi_r}{\mu_r^2} + \lambda = 0,$$

which implies $\mu_r = \sqrt{\xi_r/\lambda}$. Enforcing $\sum_r \mu_r = 1$ yields

$$\sqrt{\lambda} = \sum_{q=1}^m \sqrt{\xi_q},$$

and substitution gives (13). $\square$

4.3. Update of $\mathbf{U}$

With $\{\mathbf{H}_r\}$ and $\boldsymbol{\mu}$ fixed, the $\mathbf{U}$ -subproblem is

$$\min_{\mathbf{U} \geq 0} \text{tr}(\mathbf{U}^\top \mathbf{A} \mathbf{U}) + 2 \text{tr}(\mathbf{B}^\top \mathbf{U}), \quad (14)$$

where

$$\mathbf{A} = \sum_{r=1}^m \frac{1}{\mu_r} \mathbf{K}_r, \quad \mathbf{B} = - \sum_{r=1}^m \frac{1}{\mu_r} \mathbf{K}_r \mathbf{H}_r. \quad (15)$$

Because each $\mathbf{K}_r$ is positive semi-definite, $\mathbf{A}$ is symmetric positive semi-definite. Let $\mathbf{A}^+$ and $\mathbf{A}^-$ denote the positive and negative parts of $\mathbf{A}$, respectively, defined entrywise by

$$(\mathbf{A}^+)_{ij} = \max\{(\mathbf{A})_{ij}, 0\}, \quad (\mathbf{A}^-)_{ij} = \max\{-(\mathbf{A})_{ij}, 0\}. \quad (16)$$

Then $\mathbf{A} = \mathbf{A}^+ - \mathbf{A}^-$ and both $\mathbf{A}^+$ and $\mathbf{A}^-$ are entrywise nonnegative. A standard multiplicative update for nonnegative factorization [28] yields

$$(\mathbf{U})_{ij} \leftarrow (\mathbf{U})_{ij} \left[\frac{-(\mathbf{B})_{ij} + \sqrt{(\mathbf{B})_{ij}^2 + 4(\mathbf{A}^+ \mathbf{U})_{ij}(\mathbf{A}^- \mathbf{U})_{ij}}}{2(\mathbf{A}^+ \mathbf{U})_{ij}} \right]. \quad (17)$$

This update preserves nonnegativity by construction.

After convergence, the cluster label of sample i is obtained from the maximal component of the i th row of $\mathbf{U}$:

$$y_{ij} = \begin{cases} 1, & j = \arg \max_{j'} (\mathbf{U})_{ij'}, \\ 0, & \text{otherwise.} \end{cases} \quad (18)$$

5. Theoretical Analysis

This section records the main mathematical properties of the proposed framework. The emphasis is on mechanism-level correctness: each block update is either exact or standard, the objective is monotone, and the resulting optimization remains mathematically transparent.

5.1. Block Optimality

The $\mathbf{H}_r$ -update and μ -update are exact minimizers of their corresponding subproblems.

Proposition 2. For fixed $\mathbf{U}$ and μ , the update (10) solves (7) exactly. For fixed $\mathbf{U}$ and $\{\mathbf{H}_r\}$, the update (13) solves (12) exactly.

The proof follows from the orthogonal Procrustes theorem and the Lagrange multiplier derivation above.

5.2. Monotonic Descent

The exact block updates for H_r and μ , together with the standard descent property of the multiplicative update for U , imply monotonic decrease of the objective.

Theorem 1. Let

$$\{\mathbf{U}^{(t)}, \{\mathbf{H}_r^{(t)}\}_{r=1}^m, \mu^{(t)}\}_{t \geq 0}$$

be the sequence generated by Algorithm 1. Then the objective values satisfy

$$F(\mathbf{U}^{(t+1)}, \{\mathbf{H}_r^{(t+1)}\}, \mu^{(t+1)}) \leq F(\mathbf{U}^{(t)}, \{\mathbf{H}_r^{(t)}\}, \mu^{(t)}) \quad (19)$$

for all $t \geq 0$.

Proof. Each $\mathbf{H}_r$ update exactly minimizes the corresponding H_r subproblem with the remaining variables fixed, so it cannot increase the objective. The same is true for the update of μ . The update of $\mathbf{U}$ in (17) is a standard multiplicative rule for a nonnegative quadratic objective and satisfies the usual auxiliary-function descent property [28]. Therefore, each block update is non-increasing, and the full iteration is non-increasing as well. $\square$

5.3. Lower Boundedness and Limiting Behavior

Theorem 2. The objective in (5) is bounded below by 0. Consequently, the monotone sequence of objective values generated by Algorithm 1 converges. Moreover, if the generated iterate sequence admits an accumulation point and the corresponding block updates are well defined, then the limiting point is consistent with the optimality conditions of the closed-form $\mathbf{H}_r$ and μ updates, while the $\mathbf{U}$ block satisfies the fixed-point condition associated with the adopted multiplicative update.

Algorithm 1 Sparse Multiple Kernel Concept Factorization

Require: Original kernel matrices $\{\mathbf{K}_r^o\}_{r=1}^m$, cluster number c , neighborhood size k

Ensure: Cluster labels

```

1: for  $r = 1$  to  $m$  do
2:   Construct  $\mathbf{S}_r$  by (1)
3:   Construct the sparse kernel  $\mathbf{K}_r$  by (3)
4: end for
5: Initialize  $\mu_r = 1/m$  and initialize  $\mathbf{U} \geq 0$ 
6: repeat
7:   for  $r = 1$  to  $m$  do
8:     Update  $\mathbf{H}_r$  by (10)
9:   end for
10:  Update  $\mu$  by (13)
11:  Update  $\mathbf{U}$  by (17)
12: until convergence
13: Obtain labels by (18)

```

Proof. For each r , the quantity

$$\text{tr}(\mathbf{K}_r) - 2 \text{tr}(\mathbf{U}^\top \mathbf{K}_r \mathbf{H}_r) + \text{tr}(\mathbf{U}^\top \mathbf{K}_r \mathbf{U})$$

is a nonnegative reconstruction-type loss, and is therefore bounded below by 0. Since $\mu_r > 0$ and $\sum_{r=1}^m \mu_r = 1$, the whole objective is bounded below by 0. By the monotonicity result above, the objective sequence is non-increasing and hence convergent.

The $\mathbf{H}_r$ block and the μ block are solved exactly at each iteration. The multiplicative update for $\mathbf{U}$ is a standard nonnegative descent step that preserves feasibility. Therefore, if the produced iterate sequence admits an accumulation point and the corresponding block updates are well defined, the limiting point is consistent with the optimality conditions of the exact $\mathbf{H}_r$ and μ updates, while the $\mathbf{U}$ block satisfies the fixed-point condition of the adopted multiplicative rule. This describes the limiting behavior of the adopted blockwise scheme, but does not establish global optimality or exact block-coordinate optimality of the original nonconvex problem. $\square$

We emphasize that the multiplicative update for $\mathbf{U}$ is a descent step rather than an exact minimizer of the $\mathbf{U}$ subproblem. Therefore, the above analysis should not be interpreted as proving convergence to a global optimum or to an exact block-coordinate minimum of the nonconvex problem. As in most nonconvex matrix factorization models, different initializations may lead to different accumulation points. The experimental protocol therefore uses ten repeated runs with different random initializations and reports mean values together with standard deviations.

5.4. Complexity Analysis

The dominant cost of sparse-kernel construction is the row-wise screening of the k strongest neighbors from each original kernel matrix. In implementation, this stage consists of two parts: scanning or generating the rowwise similarities from the dense kernel source and selecting the k retained neighbors for each row. We therefore record the preprocessing complexity conservatively as

$$O(mn^2k), \quad (20)$$

which serves as an upper bound capturing neighbor screening and sparse-kernel assembly. In implementation, this preprocessing step can be carried out row by row or in blocks. Therefore, after the sparse neighbors are identified, the algorithm only needs to retain the sparse representation rather than the full dense multi-kernel collection. This is important for very large n , where storing m dense $n \times n$ kernels can be prohibitive.

At each iteration, the updates of $\{\mathbf{H}_r\}$ and $\mathbf{U}$ involve sparse–dense matrix products and thin singular value decompositions. The corresponding per-iteration cost is

$$O(mnkc + mnc^2). \quad (21)$$

Hence, the overall computational complexity is

$$O(mn^2k + (mnkc + mnc^2)t), \quad (22)$$

where t is the number of outer iterations. Once the sparse kernels are constructed, the iterative optimization part scales linearly with n for fixed m , k , and c , because the dominant operations are sparse–dense products involving $O(nk)$ nonzero entries per kernel.

For memory consumption, storing the sparse kernels requires $O(mnk)$ memory, while the factor matrices $\mathbf{U}$ and $\{\mathbf{H}_r\}$ require $O(mnc)$ memory. Therefore, the total storage complexity is

$$O(mn(k + c)). \quad (23)$$

This is significantly smaller than the $O(mn^2)$ storage cost of dense multi-kernel methods. This storage expression highlights the main large-scale advantage of the proposed framework: the memory cost depends on the number of retained neighbors rather than on the full pairwise kernel size.

6. Experiments

6.1. Experimental Design

We evaluate the proposed method on nine benchmark datasets: Trachea, Liver, Fat, MNIST4K, CBMC, USPS, TDT2, MNIST, and EMNIST. These datasets cover biological data, image data, and text data, and their sample sizes range from 1013 to 280,000. Table 2 summarizes their statistics.

Table 2. Statistics of the benchmark datasets.

Dataset	# Samples	# Features	# Classes
Trachea	1013	13,741	7
Liver	2699	7808	11
Fat	3618	15,492	9
MNIST4K	4000	784	10
CBMC	8617	1703	15
USPS	9298	256	10
TDT2	9394	36,771	30
MNIST	70,000	784	10
EMNIST	280,000	784	10

The comparison includes nine representative MKC baselines: AWP [29], OPLFMVC [6], DPMKKM [19], SimpleMKKM [3], CMKC [8], SMKC [9], M3LF [7], TFMKC [20], and SVDSMKKM [10]. Table 3 summarizes these compared methods and their main characteristics. These methods cover early-fusion, late-fusion, and scalable sampling-based MKC paradigms. Following the common practice in the MKC literature, we generate 12 base kernels for each dataset. The proposed method uses the fixed neighborhood size $k = 10$ unless otherwise stated and does not require further hyperparameter tuning. All experiments are implemented in MATLAB R2024b and executed on a workstation equipped with an Intel Core i7 CPU and 64 GB RAM.

Table 3. Compared methods and their characteristics.

Method	Category	Main Characteristic
AWP	Late fusion	Procrustes-based multiview fusion
OPLFMVC	Late fusion	One-pass late-fusion partition learning
DPMKKM	Early fusion	Discrete parameter-free MKC
SimpleMKKM	Early fusion	Hyperparameter-free kernel k -means
CMKC	Early fusion	Consistency-based sampling strategy
SMKC	Early fusion	Expectation-based scalable clustering
M3LF	Late fusion	Min-max late-fusion optimization
TFMKC	Late fusion	Tuning-free partition fusion
SVDSMKKM	Early fusion	SVD-accelerated scalable MKC

Clustering performance is evaluated using clustering accuracy (ACC), normalized mutual information (NMI), and adjusted Rand index (ARI). To reduce the influence of

random initialization, each experiment is repeated ten times with different random seeds. The reported ACC, NMI, and ARI values are the mean results over the ten repeated runs, and the corresponding standard deviations are also provided to indicate the stability of each method. We also record the wall-clock runtime. The experiments are designed not only to compare clustering quality, but also to assess whether the proposed factorization framework remains computationally feasible on large-scale datasets.

6.2. Main Quantitative Results

Tables 4–7 report the main quantitative results. The ACC, NMI, and ARI entries are reported as mean $\pm$ standard deviation over ten repeated runs. The proposed method achieves the best mean ACC, NMI, and ARI on all nine datasets in this comparison. This consistency is notable because the datasets cover substantially different sample sizes, feature dimensions, and class cardinalities. In particular, the proposed framework remains effective not only on small and medium-sized datasets such as Trachea, Liver, and Fat, but also on the much larger MNIST and EMNIST datasets. The standard deviations indicate that the random variability differs across datasets and methods. Nevertheless, the proposed method remains consistently strong in terms of the mean values of all reported clustering metrics, and reporting the standard deviations makes the comparison more transparent than reporting mean values alone.

Table 4. Clustering performance in terms of ACC. Values are reported as mean and standard deviation over ten repeated runs. The best mean result on each dataset is highlighted in bold.

Method	Trachea	Liver	Fat	MNIST4K	CBMC	USPS	TDT2	MNIST	EMNIST
AWP	0.5686 ± 0.0000	0.5432 ± 0.0000	0.5744 ± 0.0000	0.5635 ± 0.0000	0.4455 ± 0.0000	0.6355 ± 0.0000	0.5358 ± 0.0000	–	–
OPLFMVC	0.5725 ± 0.0105	0.5068 ± 0.0381	0.5462 ± 0.0297	0.5584 ± 0.0035	0.4371 ± 0.0069	0.6266 ± 0.0286	0.5146 ± 0.0117	–	–
DPMKKM	0.4002 ± 0.0033	0.5306 ± 0.0374	0.4240 ± 0.0088	0.3879 ± 0.0252	0.5397 ± 0.0168	0.3883 ± 0.0144	0.3922 ± 0.0141	–	–
SimpleMKKM	0.6111 ± 0.0000	0.3454 ± 0.0172	0.5436 ± 0.0024	0.5060 ± 0.0206	0.4696 ± 0.0167	0.5421 ± 0.0006	0.4117 ± 0.0171	–	–
CMKC	0.5994 ± 0.0516	0.4128 ± 0.0364	0.5422 ± 0.0271	0.5085 ± 0.0374	0.4324 ± 0.0251	0.5542 ± 0.0337	0.3632 ± 0.0190	0.4964 ± 0.0409	0.4718 ± 0.0162
SMKC	0.5383 ± 0.0205	0.3842 ± 0.0357	0.5738 ± 0.0227	0.5323 ± 0.0122	0.4058 ± 0.0217	0.6357 ± 0.0265	0.4360 ± 0.0219	0.5426 ± 0.0061	0.5119 ± 0.0078
M3LF	0.5623 ± 0.0283	0.4169 ± 0.0416	0.5459 ± 0.0004	0.5498 ± 0.0008	0.3946 ± 0.0134	0.6367 ± 0.0010	0.4715 ± 0.0170	–	–
TFMKC	0.5676 ± 0.0000	0.4124 ± 0.0291	0.5535 ± 0.0006	0.5384 ± 0.0002	0.4004 ± 0.0163	0.5193 ± 0.0002	0.4038 ± 0.0175	–	–
SVDSMKKM	0.5928 ± 0.0361	0.3698 ± 0.0318	0.5422 ± 0.0154	0.4977 ± 0.0403	0.4636 ± 0.0236	0.5293 ± 0.0239	0.3790 ± 0.0172	0.4913 ± 0.0378	–
Proposed	0.6721 ± 0.0657	0.5736 ± 0.0465	0.7192 ± 0.0366	0.6178 ± 0.0518	0.5489 ± 0.0309	0.7078 ± 0.0485	0.6812 ± 0.0484	0.6769 ± 0.0659	0.6408 ± 0.0392

Table 5. Clustering performance in terms of NMI. Values are reported as mean and standard deviation over ten repeated runs. The best mean result on each dataset is highlighted in bold.

Method	Trachea	Liver	Fat	MNIST4K	CBMC	USPS	TDT2	MNIST	EMNIST
AWP	0.5413 ± 0.0000	0.4623 ± 0.0000	0.5292 ± 0.0000	0.4826 ± 0.0000	0.5092 ± 0.0000	0.5793 ± 0.0000	0.6084 ± 0.0000	–	–
OPLFMVC	0.5390 ± 0.0071	0.4443 ± 0.0210	0.5252 ± 0.0139	0.4741 ± 0.0047	0.5142 ± 0.0061	0.5762 ± 0.0097	0.6061 ± 0.0049	–	–
DPMKKM	0.3865 ± 0.0090	0.3747 ± 0.0397	0.3572 ± 0.0149	0.3473 ± 0.0189	0.5065 ± 0.0151	0.3999 ± 0.0113	0.3522 ± 0.0080	–	–

Table 5. Cont.

Method	Trachea	Liver	Fat	MNIST4K	CBMC	USPS	TDT2	MNIST	EMNIST
SimpleMKKM	0.5752 ±0.0016	0.4048 ±0.0143	0.4984 ±0.0005	0.4810 ±0.0015	0.5532 ±0.0083	0.5751 ±0.0004	0.5417 ±0.0090	–	–
CMKC	0.5587 ±0.0276	0.4183 ±0.0196	0.4935 ±0.0195	0.4604 ±0.0239	0.5378 ±0.0164	0.5789 ±0.0179	0.5136 ±0.0155	0.4839 ±0.0214	0.4286 ±0.0180
SMKC	0.5324 ±0.0122	0.4000 ±0.0157	0.5119 ±0.0203	0.4604 ±0.0106	0.5027 ±0.0119	0.5904 ±0.0072	0.5641 ±0.0121	0.4917 ±0.0032	0.4516 ±0.0060
M3LF	0.5525 ±0.0022	0.4136 ±0.0263	0.5309 ±0.0005	0.4852 ±0.0006	0.4915 ±0.0097	0.5893 ±0.0010	0.6000 ±0.0072	–	–
TFMKC	0.5683 ±0.0003	0.4232 ±0.0194	0.5327 ±0.0005	0.4686 ±0.0004	0.5105 ±0.0111	0.5317 ±0.0004	0.5031 ±0.0041	–	–
SVDSMKKM	0.5580 ±0.0244	0.4048 ±0.0131	0.4989 ±0.0091	0.4658 ±0.0299	0.5453 ±0.0214	0.5572 ±0.0134	0.5222 ±0.0103	0.4879 ±0.0226	–
Proposed	0.6524 ±0.0440	0.4646 ±0.0345	0.6673 ±0.0199	0.6276 ±0.0359	0.6591 ±0.0102	0.7537 ±0.0226	0.7187 ±0.0252	0.6990 ±0.0396	0.6309 ±0.0293

Table 6. Clustering performance in terms of ARI. Values are reported as mean and standard deviation over ten repeated runs. The best mean result on each dataset is highlighted in bold.

Method	Trachea	Liver	Fat	MNIST4K	CBMC	USPS	TDT2	MNIST	EMNIST
AWP	0.3834 ±0.0000	0.2976 ±0.0000	0.4018 ±0.0000	0.3874 ±0.0000	0.3190 ±0.0000	0.4875 ±0.0000	0.3594 ±0.0000	–	–
OPLFMVC	0.3848 ±0.0097	0.2682 ±0.0346	0.3683 ±0.0232	0.3781 ±0.0040	0.3151 ±0.0035	0.4896 ±0.0094	0.3496 ±0.0062	–	–
DPMKKM	0.1649 ±0.0136	0.1252 ±0.0676	0.1466 ±0.0090	0.0843 ±0.0065	0.2772 ±0.0174	0.1139 ±0.0072	0.0509 ±0.0060	–	–
SimpleMKKM	0.4645 ±0.0011	0.1827 ±0.0170	0.3819 ±0.0006	0.3408 ±0.0083	0.3505 ±0.0100	0.4165 ±0.0004	0.2730 ±0.0221	–	–
CMKC	0.4618 ±0.0488	0.2204 ±0.0285	0.3600 ±0.0252	0.3304 ±0.0331	0.3301 ±0.0232	0.4361 ±0.0232	0.2376 ±0.0222	0.3383 ±0.0278	0.2949 ±0.0196
SMKC	0.3760 ±0.0205	0.1854 ±0.0215	0.3769 ±0.0263	0.3485 ±0.0117	0.3011 ±0.0159	0.4938 ±0.0202	0.3010 ±0.0162	0.3737 ±0.0029	0.3371 ±0.0058
M3LF	0.3973 ±0.0277	0.1993 ±0.0376	0.4103 ±0.0006	0.3719 ±0.0009	0.2910 ±0.0100	0.4852 ±0.0013	0.3258 ±0.0117	–	–
TFMKC	0.3687 ±0.0003	0.2105 ±0.0215	0.3821 ±0.0007	0.3579 ±0.0006	0.2864 ±0.0108	0.4049 ±0.0003	0.2757 ±0.0161	–	–
SVDSMKKM	0.4488 ±0.0414	0.1938 ±0.0218	0.3722 ±0.0151	0.3276 ±0.0324	0.3444 ±0.0121	0.4074 ±0.0190	0.2528 ±0.0183	0.3434 ±0.0263	–
Proposed	0.5234 ±0.0591	0.3528 ±0.0643	0.6242 ±0.0368	0.4907 ±0.0571	0.4596 ±0.0237	0.6495 ±0.0404	0.5783 ±0.0643	0.5768 ±0.0572	0.5049 ±0.0403

Table 7. Runtime comparison in seconds.

Method	Trachea	Liver	Fat	MNIST4K	CBMC	USPS	TDT2	MNIST	EMNIST
AWP	0.1961	2.1619	3.9483	5.2201	31.0314	28.9253	58.8785	N/A	N/A
OPLFMVC	0.3186	2.5821	4.6490	6.2460	35.6708	33.1717	66.7465	N/A	N/A
DPMKKM	0.2406	2.0136	3.9635	4.6558	17.0912	20.0579	38.2733	N/A	N/A
SimpleMKKM	47.0788	261.8712	206.4271	313.9156	5041.9896	1949.6581	9868.8254	N/A	N/A
CMKC	0.3429	1.5837	0.9288	1.5396	2.8598	2.3423	6.9284	23.1417	116.6071
SMKC	0.4546	0.7987	2.2072	1.6512	16.6124	15.3805	23.2306	215.5552	2149.2800
M3LF	0.9573	3.9684	6.5790	9.7054	64.1758	41.2102	137.6067	N/A	N/A
TFMKC	2.0362	37.5580	44.6000	50.0924	384.6019	227.6057	1025.8773	N/A	N/A
SVDSMKKM	3.9007	9.0990	6.0763	12.1724	16.1473	14.7079	44.5481	62.3159	N/A
Proposed	0.1350	0.5683	0.6323	0.6166	2.3277	1.7410	6.5388	22.3407	110.5469

The runtime results in Table 7 reveal a complementary advantage. The proposed method is the fastest method on all datasets in this comparison. The benefit becomes

more visible when compared with dense early-fusion methods such as SimpleMKKM and TFMKC, whose runtime grows rapidly with the dataset size. It is also noteworthy that several compared methods cannot be executed on MNIST or EMNIST because of memory limitations, whereas the proposed sparse-kernel factorization remains feasible.

6.3. Efficiency and Large-Scale Feasibility

The large-scale datasets MNIST and EMNIST are particularly informative. On MNIST, the proposed method reaches $\text{ACC} = 0.6769$ and $\text{NMI} = 0.6990$ on average while finishing in 22.34 s. On EMNIST, it reaches $\text{ACC} = 0.6408$ and $\text{NMI} = 0.6309$ on average, with runtime 110.55 s. In contrast, several dense-kernel baselines fail to run on these datasets, and even the scalable competitors require noticeably more time. This behavior is consistent with the design of the proposed model: the sparse kernels reduce memory usage, and the factorization avoids repeated dense eigendecompositions.

The memory advantage is also practically meaningful. On EMNIST, the proposed method only requires sparse-kernel storage together with the factor matrices, whereas some scalable competitors still require much larger working memory. This observation is consistent with the linear-in- n storage expression in (23) for fixed m , k , and c .

6.4. Influence of the Neighborhood Size

The neighborhood size k directly determines the sparsity pattern of the constructed kernels. A very small k may discard useful neighborhood information and may be insufficient for preserving local manifold structure, whereas an overly large k may introduce weak or noisy connections and increase memory and computational cost. Therefore, in the main experiments, $k = 10$ is used as a moderate default rather than a dataset-specific tuned parameter.

To study its influence, we test $k \in \{5, 10, 15, 20\}$ on representative datasets and report the results in Table 8. The results show that the proposed method is reasonably stable over this range. Although the best value of k may vary across datasets, $k = 10$ already yields competitive performance in most cases. This suggests that the proposed method does not rely on delicate tuning of the sparsification parameter. From the model-design perspective, increasing k can preserve more local connections, but it also reduces sparsity and increases the cost of kernel storage and sparse–dense multiplication.

Table 8. Effect of the neighborhood size k .

Dataset	$k = 5$		$k = 10$		$k = 15$		$k = 20$	
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
Trachea	0.618	0.569	0.671	0.662	0.689	0.679	0.672	0.671
Fat	0.647	0.590	0.700	0.655	0.701	0.659	0.728	0.683
MNIST4K	0.585	0.547	0.611	0.603	0.614	0.613	0.618	0.616
USPS	0.681	0.665	0.733	0.744	0.722	0.741	0.709	0.752
MNIST	0.624	0.570	0.698	0.672	0.720	0.700	0.751	0.729
EMNIST	0.533	0.468	0.635	0.598	0.671	0.641	0.681	0.664

6.5. Convergence Behavior

We further examined the convergence behavior of the proposed algorithm by tracking the objective value across iterations. For each of the nine benchmark datasets, the proposed method was repeated ten times with different random initializations, and the objective values were recorded for 100 iterations. All repeated runs showed a non-increasing objective trend. For visual clarity, Figure 1 reports one representative run on each dataset. The objective value decreases monotonically and then stabilizes after a relatively small number of iterations, which is consistent with the monotonic descent result in Theorem 1.

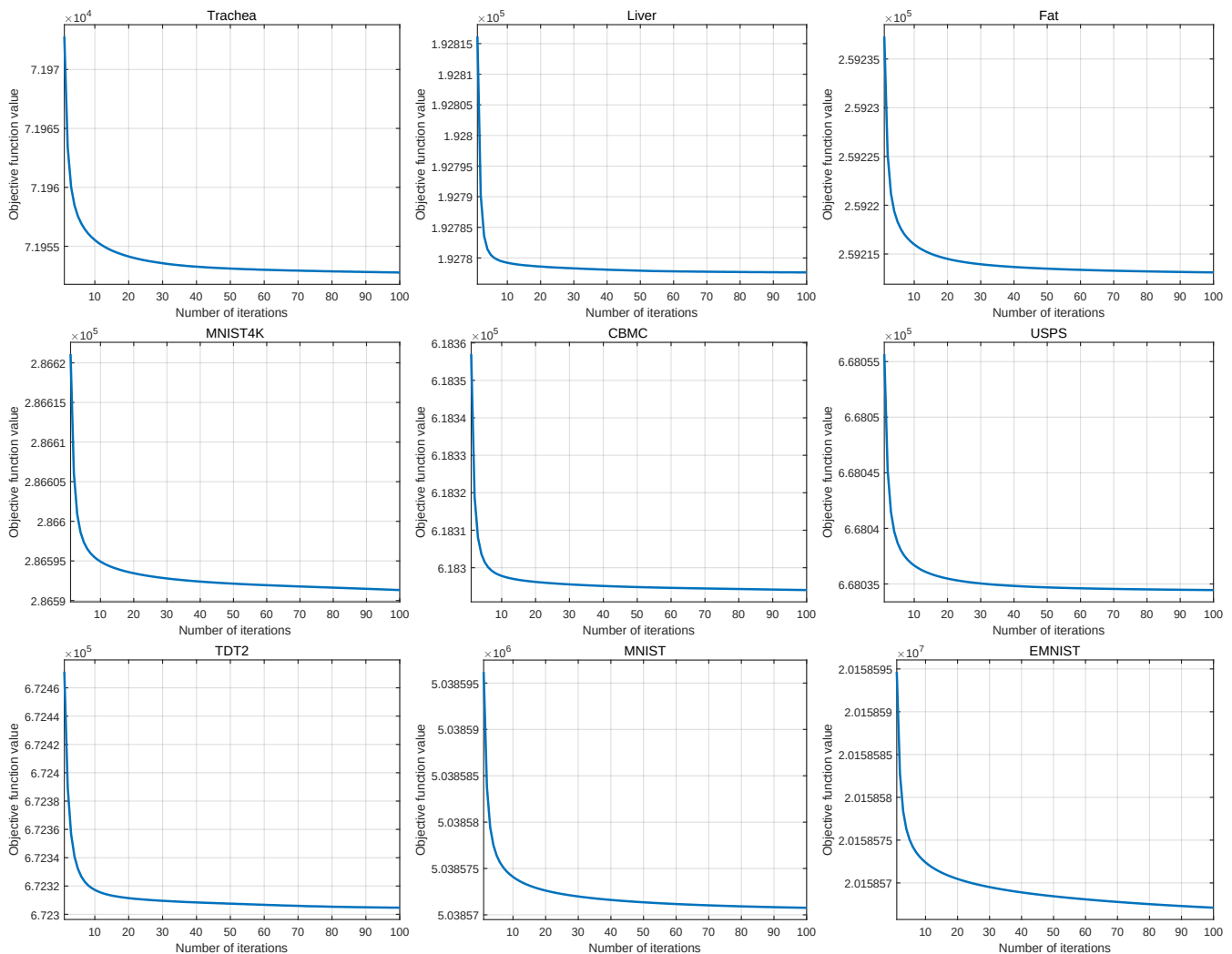


Figure 1. Convergence curves of the proposed EMKCF algorithm on nine benchmark datasets. Each panel reports one representative run, and the dataset name is shown in the panel title. The objective value was recorded for 100 iterations. The monotone decrease is consistent with the theoretical descent analysis in Theorem 1.

6.6. Experimental Discussion

The experiments support two main observations. First, the sparse-kernel construction provides a useful compromise between clustering quality and computational efficiency: it reduces storage while preserving informative local relations. Second, the shared nonnegative consensus factor, kernel-specific orthogonal factors, and adaptive kernel weights work together to produce stable clustering results across heterogeneous kernels. The reported standard deviations and convergence curves further improve the transparency of the empirical evaluation. These empirical findings are consistent with the model–optimization perspective developed in Sections 3–5

6.7. Current Scope and Limitations

The paper is deliberately positioned at the model-and-optimization level. It does not attempt to develop a statistical generalization theory for concept factorization, nor does it claim that the sparse-kernel construction is universally optimal for every multiple-kernel problem. In particular, the current implementation uses a fixed neighborhood size for simplicity and reproducibility; adaptive or learned sparsification may further improve the locality–sparsity trade-off discussed in the neighborhood-size analysis. Likewise, the con-

vergence result is stated at the level of monotonic descent, lower boundedness, objective-value convergence, and fixed-point behavior of the adopted blockwise scheme rather than global optimality or exact block-coordinate optimality. These are acceptable limitations for the current paper because the main contribution lies in the coherent mathematical construction of the model and the transparency of its optimization procedure.

The use of a shared consensus factor $\mathbf{U}$ assumes that the candidate kernels contain at least partially compatible clustering information. For a kernel with a larger reconstruction loss ξ_r , the closed-form update assigns a larger μ_r , and hence a smaller effective coefficient $1/\mu_r$. Equivalently, after eliminating μ , the weighted loss takes the form

$$\left(\sum_{r=1}^m \sqrt{\xi_r} \right)^2, \quad (24)$$

so the contribution of a large-loss kernel grows sublinearly with respect to ξ_r rather than linearly. This mechanism can reduce, but cannot completely eliminate, the influence of a poor kernel. If some kernels are dominated by severe noise or encode clustering structures that are strongly incompatible with the majority of kernels, a single shared consensus representation may still be restrictive. Developing explicit noisy-kernel detection, robust kernel selection, or multiple-consensus factorization is therefore an important direction for future work.

7. Conclusions

In this paper, we proposed a sparse multiple-kernel concept factorization framework with adaptive orthogonal factors. The method starts from a localized sparse-kernel construction that preserves positive semi-definiteness while significantly reducing storage. On top of these kernels, we formulated a weighted multiple-kernel orthogonal concept factorization model with a shared nonnegative consensus representation and kernel-specific orthogonal factors. The resulting optimization scheme is simple and structured: the orthogonal factors and the kernel weights admit closed-form updates, and the consensus representation is updated by a standard multiplicative rule. We analyzed the validity of the sparse kernels, the optimality of the closed-form subproblems, the descent behavior of the multiplicative block, the monotonicity and lower boundedness of the overall objective, and the computational complexity of the full framework. The experimental results on nine benchmark datasets further showed that the proposed method combines competitive clustering performance with attractive runtime and memory behavior. Taken together, the proposed framework forms a coherent model, optimization, and verification pipeline for sparse multiple-kernel concept factorization. Future work may consider richer structured consensus representations, dedicated mechanism ablations, sharper auxiliary-function analysis for the multiplicative block, adaptive or learned neighborhood construction, and robust kernel selection mechanisms for noisy or incompatible kernels.

Author Contributions: Conceptualization, Q.S. and L.D.; methodology, Q.S. and Y.C.; software, Y.L.; validation Y.L.; formal analysis, Y.C. and L.D.; investigation, Y.L.; data curation, Y.L.; writing—original draft preparation, Q.S.; writing—review and editing, L.D.; supervision, L.D.; funding acquisition, L.D. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China under Grants 62376146 and U2541227, by the Fundamental Research Program of Shanxi Province under Grant No. 202503021212020, and by the Open Project Foundation of the Key Laboratory of Computation Intelligence and Chinese Information Processing, Ministry of Education, under Grant No. CICIP2025001.

Data Availability Statement: The data presented in this study are openly available in [gitee] at [<https://gitee.com/csliangdu/emkcf/>] (accessed on 20 May 2026).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zhao, B.; Kwok, J.T.; Zhang, C. Multiple Kernel Clustering. In Proceedings of the 2009 SIAM International Conference on Data Mining, Philadelphia, PA, USA, 30 April–2 May. 2009; pp. 638–649.
2. Lu, Y.; Wang, L.; Lu, J.; Yang, J.; Shen, C. Multiple Kernel Clustering Based on Centered Kernel Alignment. *Pattern Recognit.* **2014**, *47*, 3656–3664.
3. Liu, X. SimpleMKKM: Simple Multiple Kernel K-Means. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 5174–5186.
4. Liu, X.; Dou, Y.; Yin, J.; Wang, L.; Zhu, E. Multiple Kernel K-Means Clustering with Matrix-Induced Regularization. In Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; pp. 1888–1894.
5. Wang, S.; Liu, X.; Zhu, E.; Tang, C.; Liu, J.; Hu, J.; Xia, J.; Yin, J. Multi-View Clustering via Late Fusion Alignment Maximization. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, Macao, China, 10–16 August 2019; pp. 3778–3784.
6. Liu, X.; Liu, L.; Liao, Q.; Wang, S.; Zhang, Y.; Tu, W.; Tang, C.; Liu, J.; Zhu, E. One Pass Late Fusion Multi-View Clustering. In *Proceedings of the 38th International Conference on Machine Learning*; Proceedings of Machine Learning Research: New York, NY, USA, 2021; Volume 139, pp. 6850–6859.
7. Li, M.; Liu, X.; Zhang, Y.; Liang, W. Late Fusion Multiview Clustering via Min–Max Optimization. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *35*, 9417–9427.
8. Liang, W.; Liu, X.; Liu, Y.; Ma, C.; Zhao, Y.; Liu, Z.; Zhu, E. Consistency of Multiple Kernel Clustering. In *Proceedings of the 40th International Conference on Machine Learning*; Proceedings of Machine Learning Research: New York, NY, USA, 2023; Volume 202, pp. 20650–20676.
9. Liang, W.; Zhu, E.; Yu, S.; Xu, H.; Zhu, X.; Liu, X. Scalable Multiple Kernel Clustering: Learning Clustering Structure from Expectation. In *Proceedings of the 41st International Conference on Machine Learning*; Proceedings of Machine Learning Research: New York, NY, USA, 2024; Volume 235, pp. 29700–29719.
10. Liang, W.; Tang, C.; Liu, X.; Liu, Y.; Liu, J.; Zhu, E.; He, K. On the Consistency and Large-Scale Extension of Multiple Kernel Clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* **2024**, *46*, 6935–6947.
11. Xu, W.; Gong, Y. Document Clustering by Concept Factorization. In Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Sheffield, UK, 25–29 July 2004; pp. 202–209.
12. Yang, B.; Zhang, X.; Nie, F.; Chen, B.; Wang, F.; Nan, Z.; Zheng, N. ECCA: Efficient Correntropy-Based Clustering Algorithm with Orthogonal Concept Factorization. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *34*, 7377–7390.
13. Mu, J.; Song, P.; Liu, X.; Li, S. Dual-Graph Regularized Concept Factorization for Multi-View Clustering. *Expert Syst. Appl.* **2023**, *223*, 119949.
14. Wu, J.; Yang, B.; Xue, Z.; Zhang, X.; Lin, Z.; Chen, B. Fast Multi-View Clustering via Correntropy-Based Orthogonal Concept Factorization. *Neural Netw.* **2024**, *173*, 106170.
15. Tao, Y.; Che, H.; Li, C.; Pan, B.; Leung, M.F. Weight Consistency and Cluster Diversity Based Concept Factorization for Multi-View Clustering. *Digit. Signal Process.* **2025**, *157*, 104879.
16. Deng, P.; Zhou, X.; Xu, J.; Huang, W.; Wang, J.; Wang, D.; Li, T. Symmetric Non-Negative Matrix Factorization-Based Deep Representation Algorithm for Multi-View Clustering. *Eng. Appl. Artif. Intell.* **2025**, *162*, 112738.
17. Khalafaoui, Y.; Matei, B.; Lovisetto, M.; Grozavu, N. Deep Matrix Factorization with Adaptive Weights for Multi-View Clustering. *Pattern Recognit.* **2026**, *170*, 112027.
18. Zhao, Y.; Hu, J.; Yang, Y.; Chang, S. Multi-View Clustering via Multiple Kernel Concept Factorization. In Proceedings of the Developments of Artificial Intelligence Technologies in Computation and Robotics: Proceedings of the 14th International FLINS Conference, Cologne, Germany, 18–21 August 2020; pp. 3–10.
19. Wang, R.; Lu, J.; Lu, Y.; Nie, F.; Li, X. Discrete and Parameter-Free Multiple Kernel K-Means. *IEEE Trans. Image Process.* **2022**, *31*, 2796–2808.
20. Zhang, J.; Li, L.; Zhang, P.; Liu, Y.; Wang, S.; Zhou, C.; Liu, X.; Zhu, E. Tuning-Free Multiple Kernel Clustering Coupled with Diverse Partition Fusion. *IEEE Trans. Neural Netw. Learn. Syst.* **2025**, *36*, 9592–9605.
21. Jiang, H.; Guo, Y.; Chen, Y.; Du, L. Late Fusion Multiple Kernel Clustering Refined via Optimal Linear Graph Filtering. In *Lecture Notes in Computer Science, Proceedings of the Machine Learning and Knowledge Discovery in Databases. Research Track, Porto, Portugal, 15–19 September 2025*; Springer: Cham, Switzerland, 2026; Volume 16013, pp. 463–479. https://doi.org/10.1007/978-3-032-05962-8_27.

22. Chen, Y.; Jiang, B.; Zhou, P.; Duan, L.; Qian, Y.; Du, L. Balanced Multiple Kernel Clustering with Discrete Partition Entropy Auto Regularization. In *Proceedings of the 33rd ACM International Conference on Multimedia*; ACM: New York, NY, USA, 2025. <https://doi.org/10.1145/3746027.3755784>.
23. Xu, J.; Duan, L.; Wang, X.; Du, L. Topological-aware Multiple Kernel Learning for Ensemble Clustering. *Pattern Recognit.* **2026**, *179*, 113588. <https://doi.org/10.1016/j.patcog.2026.113588>.
24. Wang, J.; Liu, Q.; Cai, M.; Ding, W. Neural Connected Kernel Based Multiple Kernel Clustering. *Inf. Fusion* **2026**, *125*, 103460.
25. Du, L.; Shi, Y.; Chen, Y.; Li, F.; Zhou, P.; Duan, L.; Qian, Y. A Dual Mixture-of-Experts Framework for Multi-View K-Means Clustering with View Balance and Regional Sparsity. *Inf. Fusion* **2026**, *125*, 103449. <https://doi.org/10.1016/j.inffus.2025.103449>.
26. Nadaraya, E.A. On Estimating Regression. *Theory Probab. Its Appl.* **1964**, *9*, 141–142.
27. Watson, G.S. Smooth Regression Analysis. *Sankhya Indian J. Stat. Ser. A* **1964**, *26*, 359–372.
28. Lee, D.D.; Seung, H.S. Algorithms for Non-Negative Matrix Factorization. In *Proceedings of the Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2001; Volume 13, pp. 556–562.
29. Nie, F.; Tian, L.; Li, X. Multiview Clustering via Adaptively Weighted Procrustes. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, London, UK, 19–23 August 2018; pp. 2022–2030.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.