# EFFICIENT AND EFFECTIVE MULTIPLE KERNEL CLUSTERING VIA RANDOMIZED KERNEL COMPRESSION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Multiple kernel clustering (MKC) has attracted widespread attention in recent years. Its core advantage lies in its ability to effectively capture the complex nonlinear and complementary structures of data by adaptively combining multiple kernel functions. Although various MKC methods have achieved significant success, they generally rely on the storage and spectral decomposition of multiple $n \times n$ kernel matrices. This computational bottleneck limits the practical applicability of these methods when handling large-scale datasets. To address this, we propose a scalable multiple-kernel clustering framework based on symmetric randomized compression. Specifically, by compressing each base kernel into a low-dimensional core space, we retain its dominant spectral information while substantially reducing the cost of subsequent kernel fusion and spectral optimization. Furthermore, we construct a shared subspace to capture spectral structures consistently supported across multiple kernels, enabling kernel fusion and weight learning in a common representation space. Theoretically, we establish a reconstruction-error bound for each compressed base kernel and analyze how the shared subspace captures cross-kernel consensus structures. Experimental results on eight multi-view benchmark datasets show that the proposed method achieves significant improvements in clustering performance and computational efficiency, and exhibits notable scalability advantages in large-scale data scenarios.

## 1 INTRODUCTION

Clustering is a fundamental task in unsupervised learning and has been widely used in computer vision, pattern recognition, and data mining Wang et al. (2025); Liu et al. (2023); Du et al. (2026); Wen et al. (2025); Liu et al. (2026; 2025b). Kernel clustering handles nonlinear data structures by mapping samples into a high-dimensional feature space. Since a single kernel is often limited by its kernel type and parameter settings, multiple kernel clustering (MKC) combines several candidate kernels to capture complementary information and obtain a more robust representation Liu et al. (2019; 2020); Yao et al. (2020); Ren et al. (2021); Wang et al. (2023); Nie et al. (2023); Ji & Feng (2025); Liang et al. (2025).

Despite its effectiveness, the scalability of MKC remains a major challenge. For $n$ samples, most MKC methods need to construct and store multiple $n \times n$ kernel matrices and repeatedly perform spectral decomposition or other large-scale matrix operations during optimization. These operations become expensive as $n$ grows, with direct eigendecomposition requiring up to $O(n^3)$ time. Therefore, reducing the computational complexity while preserving the main clustering structure is a key challenge for scaling MKC to large datasets. Existing scalable MKC methods often use Nyström approximation or anchor-based learning to construct compact kernel representations from a small set of representative points Liang et al. (2024; 2023); Feng et al. (2025). Although these methods reduce computation and memory, their approximation quality depends on the number and quality of the selected points. Too few points may lose important structure, while more points increase the computational cost. This motivates a natural question **: Can we move beyond sample-level approximation and directly exploit the dominant spectral structure of kernel matrices to achieve substantial compression in the spectral space while preserving key spectral information, thereby obtaining a compact kernel representation?**



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

This idea is particularly well suited to SimpleMKKM Liu (2023), which learns kernel weights and the clustering subspace through a simple min-max formulation. Its optimization, however, still relies on full $n \times n$ kernels and repeatedly computes the spectral subspace of the fused kernel. Therefore, instead of approximating the full kernel matrix using a small set of representative samples, we directly compress the dominant eigenspaces that are most relevant to the optimization and perform the subsequent weight learning and spectral optimization in a low-dimensional space. Furthermore, although different base kernels describe sample similarities in different ways, their dominant eigenspaces often contain consistent clustering information. We therefore extract this cross-kernel consistency and construct a unified low-dimensional representation for multiple-kernel fusion.

Following this motivation, we develop a randomized symmetric kernel compression framework for SimpleMKKM with two compression strategies. Independent compression captures the dominant eigenspace of each base kernel separately and constructs a compact core representation, while shared compression jointly aggregates the dominant spectral subspaces of multiple base kernels to extract their shared structure and construct a common low-dimensional space. As a result, kernel-weight learning and spectral optimization in SimpleMKKM can be moved from the original $n \times n$ kernel space to a low-dimensional core space, reducing the cost of repeated spectral computation.

The main contributions of this work are summarized as follows.

- We propose a scalable SimpleMKKM framework based on randomized symmetric kernel compression. By extracting the dominant low-dimensional spectral information of each base kernel, the repeated spectral optimization in the original kernel space is moved to a much smaller core space. We further develop a shared-subspace compression method to capture cross-kernel consensus structure and construct a unified low-dimensional representation for kernel fusion and weight learning.
- We theoretically analyze the effect of randomized compression on the base kernels and establish a reconstruction-error bound for the compressed representation. We further analyze the ability of the shared subspace to represent cross-kernel consensus structure.
- We conduct extensive experiments on multi-view benchmark datasets. The results show that the proposed method maintains competitive clustering performance while greatly reducing the running time, and achieves good scalability on large-scale datasets.

## 2 Related Work

### 2.1 Multiple Kernel Clustering

Let $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n \subset \mathbb{R}^d$ denote a collection of $n$ samples. For the $p$-th base kernel, a feature mapping $\phi_p : \mathcal{X} \to \mathcal{H}_p$ embeds each sample into a reproducing kernel Hilbert space $\mathcal{H}_p$, where $1 \le p \le m$. Given nonnegative weights $\boldsymbol{\gamma} = [\gamma_1, \ldots, \gamma_m]^\top$, the multiple-kernel representation of $\mathbf{x}$ can be written as

$$\phi_{\boldsymbol{\gamma}}(\mathbf{x}) = \left[\gamma_1\phi_1(\mathbf{x})^\top, \ldots, \gamma_m\phi_m(\mathbf{x})^\top\right]^\top. \tag{1}$$

The inner product between two such representations induces $\kappa_{\boldsymbol{\gamma}}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{p=1}^m \gamma_p^2 \kappa_p(\mathbf{x}_i, \mathbf{x}_j)$. Accordingly, the fused kernel matrix is $\mathbf{K}_{\boldsymbol{\gamma}} = \sum_{p=1}^m \gamma_p^2 \mathbf{K}_p$.

Multiple kernel $k$-means (MKKM) jointly learns the kernel weights and a consensus clustering partition (Huang et al., 2012; Yu et al., 2012). Its relaxed formulation is

$$\begin{aligned} \min_{\mathbf{H},\boldsymbol{\gamma}} \quad & \mathrm{Tr}\big(\mathbf{K}_{\boldsymbol{\gamma}}(\mathbf{I}_n - \mathbf{H}\mathbf{H}^\top)\big), \\ \text{s.t.} \quad & \mathbf{H} \in \mathbb{R}^{n\times k}, \quad \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k, \\ & \boldsymbol{\gamma} \in \mathbb{R}_+^m, \quad \mathbf{1}^\top\boldsymbol{\gamma} = 1, \end{aligned} \tag{2}$$

where $\mathbf{H}$ is the relaxed cluster-indicator matrix and $\Delta = \{\boldsymbol{\gamma} \in \mathbb{R}_+^m : \mathbf{1}^\top\boldsymbol{\gamma} = 1\}$ denotes the feasible set of kernel weights. Later variants introduce localized kernel weights or matrix-induced regularization to capture sample-specific structure and reduce redundancy among the base kernels (Gönen & Margolin, 2014; Liu et al., 2016). Nevertheless, their main optimization steps still operate on dense $n \times n$ Gram matrices and require repeated spectral computation.

### 2.2 SIMPLEMKKM

SimpleMKKM (Liu, 2023) provides a simple min–max formulation for multiple kernel clustering by replacing the conventional within-cluster scatter objective of MKKM with an unsupervised kernel-alignment criterion. It learns the kernel weights and relaxed partition through

$$\begin{aligned} \min_{\boldsymbol{\gamma}\in\Delta} \max_{\mathbf{H}} \quad & \operatorname{Tr}\left(\mathbf{K}_{\boldsymbol{\gamma}}\mathbf{H}\mathbf{H}^{\top}\right), \\ \text{s.t.} \quad & \mathbf{H}\in\mathbb{R}^{n\times k}, \quad \mathbf{H}^{\top}\mathbf{H}=\mathbf{I}_k. \end{aligned} \tag{3}$$

The inner maximization is a kernel $k$-means spectral problem. Defining its optimal value as

$$\mathcal{J}(\boldsymbol{\gamma}) = \max_{\mathbf{H}^{\top}\mathbf{H}=\mathbf{I}_k} \operatorname{Tr}\left(\mathbf{K}_{\boldsymbol{\gamma}}\mathbf{H}\mathbf{H}^{\top}\right), \tag{4}$$

transforms Problem (3) into $\min_{\gamma\in\Delta}\mathcal{J}(\boldsymbol{\gamma})$. SimpleMKKM thus reduces joint partition and kernel-weight learning to an optimal-value problem over $\boldsymbol{\gamma}$, optimized on the simplex by reduced-gradient descent without additional regularization hyperparameters. However, each objective and gradient evaluation still requires the leading eigenspace of $\mathbf{K}_{\boldsymbol{\gamma}}\in\mathbb{R}^{n\times n}$, so its spectral and memory costs grow rapidly with $n$.

## 3 PROPOSED METHOD

This section presents a scalable SimpleMKKM framework based on randomized core spaces. We construct independent and shared low-dimensional kernel representations, perform kernel-weight learning and spectral optimization using $r\times r$ cores, and then recover sample-level clustering representations. We establish a reconstruction-error bound for each compressed base kernel and analyze the structural consistency of shared compression.

### 3.1 RANDOMIZED SVD PRELIMINARIES

Randomized SVD uses Gaussian random projection to approximate the dominant column space of a matrix, avoiding a full spectral decomposition of a large matrix Halko et al. (2011). Specifically, the matrix is multiplied by a low-dimensional Gaussian random matrix, and the resulting sample matrix is orthogonalized to obtain a low-dimensional basis that captures the dominant spectral information. Inspired by this idea, we apply the same strategy to compress the kernel matrix.

**Definition 1** (Randomized kernel sampling subspace)**.** *For $\mathbf{K}=\mathbf{K}^{\top}\succeq 0$, let $r$ be the target rank, $s\geq 0$ the oversampling parameter, and $q\geq 0$ the power iteration parameter. Let $\ell=\min\{n, r+s\}$ and generate $\boldsymbol{\Omega}\in\mathbb{R}^{n\times\ell}$ with $\Omega_{ij}\overset{\text{i.i.d.}}{\sim}\mathcal{N}(0,1)$. Define*

$$\mathbf{Z} = \operatorname{orth}\left(\mathbf{K}^{q+1}\boldsymbol{\Omega}\right). \tag{5}$$

Here, $\operatorname{orth}(\cdot)$ returns an orthonormal basis for the column space of its input. If the sampled matrix has full column rank, then $\mathbf{Z}^{\top}\mathbf{Z}=\mathbf{I}_{\ell}$. Power iteration improves spectral separation while preserving the eigenvectors of $\mathbf{K}$, allowing $\mathbf{Z}$ to better capture the dominant spectral subspace. Since $\mathbf{Z}$ spans an oversampled subspace of dimension $\ell$, we further apply Rayleigh–Ritz truncation to extract its leading $r$-dimensional subspace.

**Definition 2** (Symmetric kernel compression)**.** *Given a kernel matrix $\mathbf{K}=\mathbf{K}^{\top}\succeq 0$ and its sampling basis $\mathbf{Z}$, define the compressed matrix $\mathbf{T}=\mathbf{Z}^{\top}\mathbf{K}\mathbf{Z}$. Let $\mathbf{V}_r\in\mathbb{R}^{\ell\times r}$ contain the eigenvectors associated with the largest $r$ eigenvalues of $\mathbf{T}$. Define*

$$\mathbf{Q}=\mathbf{Z}\mathbf{V}_r, \qquad \mathbf{C}=\mathbf{Q}^{\top}\mathbf{K}\mathbf{Q}, \qquad \widehat{\mathbf{K}}=\mathbf{Q}\mathbf{C}\mathbf{Q}^{\top}. \tag{6}$$

*Then*

$$\mathbf{Q}^{\top}\mathbf{Q}=\mathbf{I}_r, \qquad \widehat{\mathbf{K}}=\widehat{\mathbf{K}}^{\top}\succeq 0, \qquad \operatorname{rank}(\widehat{\mathbf{K}})\leq r.$$

**Remark 1.** *The matrix $\mathbf{Q}$ can be viewed as a spectral probe that explores the sampled subspace and identifies the most informative directions of $\mathbf{K}$. These directions form a compact basis for low-rank kernel reconstruction.*

### 3.2 FORMULATION

The main computational burden of SimpleMKKM arises from repeatedly constructing and decomposing the $n \times n$ fused kernel matrix during kernel-weight optimization. To reduce this cost, we first develop an independent randomized compression scheme for SimpleMKKM. The key idea is to extract the dominant spectral information of each base kernel separately and compress each original kernel into an $r \times r$ core matrix, allowing kernel-weight learning to be performed in a low-dimensional space. Based on Definitions 1 and 2, we consider two strategies for compressing multiple kernel matrices: independent compression and consensus compression.

**Definition 3** (Independent kernel compression)**.** *Given $m$ symmetric positive semidefinite base kernels $\{\mathbf{K}_p \in \mathbb{R}^{n\times n}\}_{p=1}^m$, let $\mathbf{Z}_p$ be the randomized sampling basis of $\mathbf{K}_p$ given by Definition 1. Applying Definition 2 to each sampled subspace gives*

$$\mathbf{T}_p = \mathbf{Z}_p^\top \mathbf{K}_p \mathbf{Z}_p, \qquad \mathbf{Q}_p = \mathbf{Z}_p \mathbf{V}_{p,r}, \tag{7}$$

*where $\mathbf{V}_{p,r}$ contains the leading $r$ eigenvectors of $\mathbf{T}_p$. The corresponding core matrix and sample-space reconstruction are*

$$\mathbf{C}_p^{(\mathrm{ind})} = \mathbf{Q}_p^\top \mathbf{K}_p \mathbf{Q}_p \in \mathbb{R}^{r\times r}, \qquad \widehat{\mathbf{K}}_p = \mathbf{Q}_p \mathbf{C}_p^{(\mathrm{ind})} \mathbf{Q}_p^\top. \tag{8}$$

We next use the compressed core matrices $\{\mathbf{C}_p^{(\mathrm{ind})}\}_{p=1}^m$ for kernel-weight learning. Let the kernel-weight vector be $\boldsymbol{\gamma} = [\gamma_1, \ldots, \gamma_m]^\top \in \Delta$, where $\Delta := \{\boldsymbol{\gamma} \in \mathbb{R}_+^m \mid \sum_{p=1}^m \gamma_p = 1\}$.

We formulate the following low-dimensional SimpleMKKM model:

$$\begin{aligned} \min_{\boldsymbol{\gamma}\in\Delta} \max_{\mathbf{H}} \quad & \mathrm{Tr}\left(\left(\sum_{p=1}^m \gamma_p^2 \mathbf{C}_p^{(\mathrm{ind})}\right)\mathbf{H}\mathbf{H}^\top\right), \\ \text{s.t.} \quad & \mathbf{H} \in \mathbb{R}^{r\times k}, \qquad \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k, \end{aligned} \tag{9}$$

Here, $\mathbf{H}$ denotes the relaxed spectral representation in the core space. This formulation follows the SimpleMKKM optimization scheme while performing kernel-weight learning in a compact core space, thereby avoiding repeated large-scale spectral computation.

### 3.3 KERNEL-WISE COMPRESSION FIDELITY

**Theorem 1** (Base-kernel compression error)**.** *Let $\lambda_{p,r+1}$ be the $(r+1)$st largest eigenvalue of $\mathbf{K}_p$. For every realized sketch, the symmetric rank-$r$ construction satisfies*

$$\|\mathbf{K}_p - \widehat{\mathbf{K}}_p\|_2 \le 2\|(\mathbf{I} - \mathbf{Z}_p\mathbf{Z}_p^\top)\mathbf{K}_p\|_2 + \lambda_{p,r+1}. \tag{10}$$

The first term measures information missed by the sampled range; the second is the spectral tail discarded by rank-$r$ truncation. Thus, reconstruction is accurate when the sketch captures the dominant eigenspace and the spectrum decays rapidly beyond rank $r$. Power iteration can improve the sampled range. Appendix A.1.1 provides the proof and the Gaussian range-finding estimate of Halko et al. (2011); Appendix A.1.2 gives the corresponding uniform bound over the base-kernel collection.

### 3.4 SHARED-SUBSPACE COMPRESSION

Independent compression constructs a separate low-dimensional basis $\mathbf{Q}_p$ for each base kernel to preserve its dominant spectral information. Since these low-dimensional representations are learned independently for different kernels, they are not explicitly aligned and may not directly capture the shared structure across kernels. To address this issue, we further propose shared-subspace compression. Specifically, we aggregate the oversampled subspaces of all base kernels and extract a shared $r$-dimensional basis. We then project all base kernels onto this shared subspace, providing a unified low-dimensional representation for core-matrix fusion and kernel-weight learning.

**Definition 4** (Consensus spectral subspace)**.** *For sampled bases $\mathbf{Z}_p$ satisfying $\mathbf{Z}_p^\top\mathbf{Z}_p = \mathbf{I}_\ell$, we seek an $r$-dimensional orthogonal subspace that has the largest total overlap with the sampled subspaces:*

$$\mathbf{Q}^\star \in \arg\max_{\mathbf{Q}^\top\mathbf{Q}=\mathbf{I}_r} \sum_{p=1}^m \|\mathbf{Q}^\top\mathbf{Z}_p\|_F^2. \tag{11}$$

**Proposition 1** (Closed-form consensus subspace)**.** *Let* $\mathbf{Z}_{\mathrm{cat}} = [\mathbf{Z}_1, \ldots, \mathbf{Z}_m] = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$. *An optimizer is* $\mathbf{Q}^\star = \mathbf{U}_{(:,1:r)}$.

For any feasible $\mathbf{Q}$,

$$\|(\mathbf{I} - \mathbf{Q}\mathbf{Q}^\top)\mathbf{Z}_p\|_F^2 = \ell - \|\mathbf{Q}^\top \mathbf{Z}_p\|_F^2. \tag{12}$$

Hence, the same $\mathbf{Q}^\star$ minimizes the aggregate projection residual of the sampled subspaces. This strategy effectively extracts cross-kernel consensus information, enhances structural consistency among different kernels, and thereby improves clustering performance. Appendix A.1.3 retains the full derivation of the closed-form solution.

Using this shared basis, define the core kernel matrices as follows.

**Proposition 2** (Fusion–compression consistency)**.** *For every* $\boldsymbol{\gamma} \in \Delta$,

$$\mathbf{C}_p^{(\mathrm{sha})} = \mathbf{Q}^\top \mathbf{K}_p \mathbf{Q}, \qquad \mathbf{C}_{\boldsymbol{\gamma}}^{(\mathrm{sha})} = \sum_p \gamma_p^2 \mathbf{C}_p^{(\mathrm{sha})} = \mathbf{Q}^\top \mathbf{K}_{\boldsymbol{\gamma}} \mathbf{Q}. \tag{13}$$

The identity follows directly from linearity: compressing the fused kernel is equivalent to fusing the compressed kernels. All base kernels therefore share one spectral coordinate system. Restricting the SimpleMKKM spectral representation to $\mathrm{range}(\mathbf{Q})$ gives

$$\begin{aligned} \min_{\boldsymbol{\gamma}\in\Delta} \max_{\mathbf{H}} \quad & \mathrm{Tr}\left(\left(\sum_{p=1}^m \gamma_p^2 \mathbf{Q}^\top \mathbf{K}_p \mathbf{Q}\right)\mathbf{H}\mathbf{H}^\top\right), \\ \text{s.t.} \quad & \mathbf{H} \in \mathbb{R}^{r\times k}, \qquad \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k, \end{aligned} \tag{14}$$

### 3.5 Preservation of the Dominant Clustering Subspace

Spectral clustering depends primarily on the leading eigenspace of the fused kernel, rather than on entrywise recovery of the entire matrix. The following condition makes this connection explicit.

**Proposition 3** (Lossless recovery under subspace containment)**.** *Fix* $\boldsymbol{\gamma} \in \Delta$, *and let* $\mathbf{U}_{\boldsymbol{\gamma},k} \in \mathbb{R}^{n\times k}$ *contain leading orthonormal eigenvectors of* $\mathbf{K}_{\boldsymbol{\gamma}}$. *If* $k \le r$ *and*

$$\mathrm{range}(\mathbf{U}_{\boldsymbol{\gamma},k}) \subseteq \mathrm{range}(\mathbf{Q}), \tag{15}$$

*then* $\mathbf{H}_{\boldsymbol{\gamma},k} = \mathbf{Q}^\top \mathbf{U}_{\boldsymbol{\gamma},k}$ *is a leading orthonormal* $k$*-dimensional eigenspace of* $\mathbf{C}_{\boldsymbol{\gamma}}^{(\mathrm{sha})}$, *and*

$$\mathbf{Q}\mathbf{H}_{\boldsymbol{\gamma},k} = \mathbf{U}_{\boldsymbol{\gamma},k}. \tag{16}$$

*If the* $k$*th and* $(k+1)$*st eigenvalues are separated, this leading subspace is unique up to an orthogonal change of basis.*

Appendix A.1.4 gives a short proof. It clarifies the role of the compression rank: increasing $r$ enlarges the shared space and makes it more likely to cover the cluster-relevant directions, while increasing the cost of the core problem.

### 3.6 OPTIMIZATION

The independent and shared compression schemes lead to the low-dimensional SimpleMKKM objectives in Eqs. (9) and (14), respectively. Although their core matrices are constructed from different bases, both objectives have the same min–max form. We therefore describe their optimization using a common notation. Let $\xi \in \{\mathrm{ind}, \mathrm{sha}\}$ denote the compression mode, corresponding to independent and shared compression, respectively. The $p$-th core kernel is defined as

$$\mathbf{C}_p^{(\xi)} = \begin{cases} \mathbf{Q}_p^\top \mathbf{K}_p \mathbf{Q}_p, & \xi = \mathrm{ind}, \\ \mathbf{Q}^\top \mathbf{K}_p \mathbf{Q}, & \xi = \mathrm{sha}, \end{cases} \qquad p = 1, \ldots, m. \tag{17}$$

For either mode, the corresponding fused core is $\mathbf{C}_{\boldsymbol{\gamma}}^{(\xi)} = \sum_{p=1}^m \gamma_p^2 \mathbf{C}_p^{(\xi)}$. Consequently, the two compressed SimpleMKKM formulations can be expressed in the unified form

$$\begin{aligned} \min_{\boldsymbol{\gamma}\in\Delta} \max_{\mathbf{H}^{(\xi)}} \quad & \mathrm{Tr}\Big(\mathbf{C}_{\boldsymbol{\gamma}}^{(\xi)} \mathbf{H}^{(\xi)} {\mathbf{H}^{(\xi)}}^\top\Big), \\ \text{s.t.} \quad & \mathbf{H}^{(\xi)} \in \mathbb{R}^{r\times k}, \qquad {\mathbf{H}^{(\xi)}}^\top \mathbf{H}^{(\xi)} = \mathbf{I}_k, \end{aligned} \qquad \xi \in \{\mathrm{ind}, \mathrm{sha}\}. \tag{18}$$

Following SimpleMKKM (Liu, 2023), we optimize the kernel weights through the optimal-value function

$$\mathcal{J}^{(\xi)}(\boldsymbol{\gamma}) = \max_{\mathbf{H}^{(\xi)\top}\mathbf{H}^{(\xi)}=\mathbf{I}_k} \operatorname{Tr}\Big(\mathbf{C}_{\boldsymbol{\gamma}}^{(\xi)}\mathbf{H}^{(\xi)}{\mathbf{H}^{(\xi)}}^{\top}\Big), \tag{19}$$

and solve

$$\min_{\boldsymbol{\gamma}\in\Delta} \mathcal{J}^{(\xi)}(\boldsymbol{\gamma}), \qquad \xi \in \{\mathrm{ind}, \mathrm{sha}\}. \tag{20}$$

The following steps are applied separately for each fixed mode $\xi$; we omit the mode superscript of $\boldsymbol{\gamma}$ for clarity.

**Initialization.** We initialize the kernel weights uniformly as $\gamma_p^{(1)} = \frac{1}{m}, \qquad p = 1, \ldots, m.$

**Solving the inner spectral problem.** At iteration $t$, we form $\mathbf{C}_{\boldsymbol{\gamma}^{(t)}}^{(\xi)}$ using the fused-core definition above. By the Rayleigh–Ritz theorem, the inner maximizer $\mathbf{H}^{(\xi,t)}$ consists of its $k$ leading eigenvectors, and $\mathcal{J}^{(\xi)}(\boldsymbol{\gamma}^{(t)})$ is the sum of the corresponding eigenvalues.

**Updating $\boldsymbol{\gamma}$.** When the $k$th and $(k+1)$st core eigenvalues are separated (or $k = r$), the optimal-value differentiation rule gives

$$g_p^{(\xi,t)} := \left.\frac{\partial \mathcal{J}^{(\xi)}(\boldsymbol{\gamma})}{\partial \gamma_p}\right|_{\boldsymbol{\gamma}=\boldsymbol{\gamma}^{(t)}} = 2\gamma_p^{(t)} \operatorname{Tr}\Big(\mathbf{C}_p^{(\xi)}\mathbf{H}^{(\xi,t)}{\mathbf{H}^{(\xi,t)}}^{\top}\Big), \qquad p = 1, \ldots, m. \tag{21}$$

For numerical scaling, we set $\bar{\mathbf{g}}^{(\xi,t)} = \mathbf{g}^{(\xi,t)}/\|\mathbf{g}^{(\xi,t)}\|_2$ when the gradient is nonzero; otherwise, the weights remain unchanged. To account for the simplex constraint, we choose the largest weight as the reference component, $u^{(t)} = \arg\max_{1\le p\le m} \gamma_p^{(t)}$, and construct the reduced gradient

$$\widetilde{g}_p^{(\xi,t)} = \begin{cases} \bar{g}_p^{(\xi,t)} - \bar{g}_{u^{(t)}}^{(\xi,t)}, & p \neq u^{(t)}, \\ \displaystyle\sum_{j\neq u^{(t)}} \Big(\bar{g}_{u^{(t)}}^{(\xi,t)} - \bar{g}_j^{(\xi,t)}\Big), & p = u^{(t)}. \end{cases} \tag{22}$$

At a zero-weight component, we suppress any step that would make the weight negative. The resulting direction is

$$d_p^{(\xi,t)} = \begin{cases} 0, & p \neq u^{(t)},\ \gamma_p^{(t)} = 0,\ \widetilde{g}_p^{(\xi,t)} > 0, \\ -\widetilde{g}_p^{(\xi,t)}, & p \neq u^{(t)} \text{ otherwise}, \\ \displaystyle -\sum_{j\neq u^{(t)}} d_j^{(\xi,t)}, & p = u^{(t)}. \end{cases} \tag{23}$$

We choose a feasible step size $\alpha_t^{(\xi)}$ by line search to decrease $\mathcal{J}^{(\xi)}$, and update the weights as

$$\boldsymbol{\gamma}^{(t+1)} = \boldsymbol{\gamma}^{(t)} + \alpha_t^{(\xi)}\mathbf{d}^{(\xi,t)}. \tag{24}$$

**Recovering the sample-level representations.** The two schemes recover sample-level representations differently: independent compression uses the reconstructed fused kernel, whereas shared-subspace compression directly lifts the core eigenvectors through the common basis $\mathbf{Q}$.

*Independent compression.* For the independently compressed model, the $p$-th base kernel is represented by its orthonormal basis $\mathbf{Q}_p$ and core matrix $\mathbf{C}_p^{(\mathrm{ind})}$ as $\mathbf{K}_p \approx \mathbf{Q}_p\mathbf{C}_p^{(\mathrm{ind})}\mathbf{Q}_p^{\top}$. Because the bases $\{\mathbf{Q}_p\}_{p=1}^m$ generally define different coordinate systems, $\mathbf{H}^{(\mathrm{ind})}$ serves as the solution of the core spectral problem used to evaluate the optimal-value function and its gradient. The sample-level representation is recovered from the fused kernel. Given the converged weights $\boldsymbol{\gamma}^\star$, this kernel is

$$\widehat{\mathbf{K}}_{\boldsymbol{\gamma}^\star}^{(\mathrm{ind})} = \sum_{p=1}^{m} (\gamma_p^\star)^2 \mathbf{Q}_p\mathbf{C}_p^{(\mathrm{ind})}\mathbf{Q}_p^{\top}. \tag{25}$$

To avoid forming the $n \times n$ matrix explicitly, we factor each positive semidefinite core as $\mathbf{C}_p^{(\mathrm{ind})} = \mathbf{R}_p\mathbf{R}_p^{\top}$ and assemble the joint low-rank factor

$$\mathbf{L}_{\boldsymbol{\gamma}^\star} = [\gamma_1^\star\mathbf{Q}_1\mathbf{R}_1, \ldots, \gamma_m^\star\mathbf{Q}_m\mathbf{R}_m] \in \mathbb{R}^{n\times mr}. \tag{26}$$

It follows that $\widehat{\mathbf{K}}^{(\mathrm{ind})}_{\boldsymbol{\gamma}^\star} = \mathbf{L}_{\boldsymbol{\gamma}^\star}\mathbf{L}_{\boldsymbol{\gamma}^\star}^\top$. Let $\mathbf{L}_{\boldsymbol{\gamma}^\star} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top$ be its singular value decomposition. The sample-level spectral representation is given by the $k$ leading left singular vectors,

$$\mathbf{F}^{(\mathrm{ind})} = \mathbf{U}_{(:,1:k)} \in \mathbb{R}^{n\times k}. \tag{27}$$

*Shared-subspace compression.* For the shared-subspace model, the converged core-space representation $\mathbf{H}^{(\mathrm{sha},\star)} \in \mathbb{R}^{r\times k}$ is expressed in the common coordinate system defined by $\mathbf{Q} \in \mathbb{R}^{n\times r}$. Its sample-level representation is therefore obtained directly as

$$\mathbf{F}^{(\mathrm{sha})} = \mathbf{Q}\mathbf{H}^{(\mathrm{sha},\star)} \in \mathbb{R}^{n\times k}. \tag{28}$$

For either compression scheme, the $i$-th row of $\mathbf{F}^{(\xi)}$ represents the $i$-th sample. We apply standard $k$-means to these row vectors to obtain the final cluster assignments.

# 4 Experiment

To evaluate the clustering performance and scalability of the proposed independent and shared-subspace compression methods, we conduct extensive experiments on several benchmark multi-view datasets and compare them with representative multiple kernel clustering and scalable kernel approximation methods.

## 4.1 Datasets and Comparison Algorithms

We evaluate the proposed methods on eight real-world benchmark multi-view datasets: Reuters Amini et al. (2009), 100Leaves Mallah et al. (2013), HW-2view Asuncion et al. (2007), Cora McCallum et al. (2000), Scene15 Lazebnik et al. (2006), CCV Jiang et al. (2011), Hdigit Asuncion et al. (2007), and ALOI100 Geusebroek et al. (2005). We compare the proposed approach with six representative multiple kernel clustering methods from the recent literature: MKKM Huang et al. (2012), RMKKM Du et al. (2015), LMKKM Gönen & Margolin (2014), LMTC Liu et al. (2025a), CMKC Liang et al. (2023) and SMKKM Liu (2023).

## 4.2 Performance comparison

Table 1 presents the ACC, NMI, Purity, and F-score results of all compared methods. Regarding ACC, the following observations can be made:

SMKKM-Sha achieves the best ACC on seven of the eight datasets. Compared with the strongest competing method on each dataset, it improves ACC by 1.81%, 1.94%, 7.25%, 8.63%, 4.42%, 13.56%, and 2.23% on 100Leaves, HW-2view, Cora, Scene15, CCV, Hdigit, and ALOI100, respectively. The performance advantage of SMKKM-Sha mainly stems from its unified representation of multi-kernel spectral structures in a shared subspace, which facilitates the effective fusion of complementary information across base kernels.

The proposed methods achieve geometric-mean speedups of $140.52\times$ and $112.68\times$ for SMKKM-Ind and SMKKM-Sha over SMKKM, respectively. The acceleration mainly comes from performing kernel fusion and spectral optimization in the compact core space, avoiding repeated large-scale eigendecompositions while maintaining competitive clustering performance.

In summary, our methods outperform the compared methods on most datasets and run nearly 100 times faster than SimpleMKKM on average. By performing kernel fusion and optimization in a compact low-dimensional space, they maintain clustering accuracy while greatly reducing computational cost, making multiple kernel clustering more practical for large-scale applications.

## 4.3 Convergence Analysis

To evaluate the convergence of the proposed algorithm, we recorded the objective value over iterations on eight datasets (Fig. 1). On all datasets, the objective value decreases monotonically without noticeable oscillation. Most of the decrease occurs in the first few iterations, and the curves generally become stable after about 10 iterations. These results show that the proposed algorithm converges quickly and stably across datasets of different sizes and characteristics.

Table 1: Clustering performance (%) on the benchmark multi-view datasets. The best and second-best results are highlighted in bold and underlined, respectively.

| **Datasets** | MKKM | RMKKM | LMKKM | LMTC | CMKC | SMKKM | **SMKKM-Ind** | **SMKKM-Sha** |
|---|---|---|---|---|---|---|---|---|
| | | | | **ACC (%)** | | | | |
| Reuters | $37.98 \pm 3.94$ | $39.69 \pm 3.06$ | $50.27 \pm 1.00$ | $50.13 \pm 0.06$ | $48.17 \pm 0.00$ | $\mathbf{50.65 \pm 0.95}$ | $49.20 \pm 1.60$ | $49.40 \pm 1.18$ |
| 100Leaves | $47.13 \pm 2.64$ | $60.94 \pm 2.28$ | $82.69 \pm 2.23$ | $68.64 \pm 1.54$ | $82.30 \pm 1.98$ | $82.66 \pm 1.56$ | $81.86 \pm 1.39$ | $\mathbf{84.50 \pm 1.10}$ |
| HW-2view | $72.75 \pm 6.56$ | $81.01 \pm 6.60$ | $76.06 \pm 0.56$ | $89.92 \pm 0.21$ | $68.60 \pm 0.42$ | $73.47 \pm 1.46$ | $69.99 \pm 0.52$ | $\mathbf{91.86 \pm 0.07}$ |
| Cora | $20.13 \pm 1.85$ | $18.89 \pm 0.66$ | $24.37 \pm 0.16$ | $41.96 \pm 1.22$ | $32.77 \pm 0.16$ | $39.33 \pm 0.41$ | $34.47 \pm 0.83$ | $\mathbf{49.21 \pm 0.17}$ |
| Scene15 | $38.47 \pm 1.26$ | $39.88 \pm 1.10$ | $40.37 \pm 1.31$ | $39.50 \pm 0.93$ | $39.05 \pm 0.76$ | $40.07 \pm 1.21$ | $39.59 \pm 0.96$ | $\mathbf{49.00 \pm 0.66}$ |
| CCV | $15.63 \pm 0.44$ | $15.83 \pm 0.57$ | $19.90 \pm 0.68$ | $22.96 \pm 0.61$ | $21.44 \pm 0.84$ | $22.04 \pm 0.74$ | $21.96 \pm 0.54$ | $\mathbf{27.38 \pm 0.56}$ |
| Hdigit | $74.47 \pm 7.33$ | $78.48 \pm 4.92$ | $72.31 \pm 1.15$ | $78.55 \pm 2.75$ | $72.48 \pm 1.23$ | $71.34 \pm 0.12$ | $69.54 \pm 0.05$ | $\mathbf{92.11 \pm 0.01}$ |
| ALOI100 | $50.78 \pm 2.95$ | $64.75 \pm 1.74$ | $75.46 \pm 1.26$ | $67.32 \pm 1.84$ | $74.90 \pm 1.99$ | $73.67 \pm 1.58$ | $74.56 \pm 1.29$ | $\mathbf{77.69 \pm 1.42}$ |
| | | | | **NMI (%)** | | | | |
| Reuters | $21.01 \pm 2.84$ | $22.65 \pm 2.60$ | $30.84 \pm 1.03$ | $\mathbf{32.07 \pm 0.06}$ | $29.59 \pm 0.00$ | $30.23 \pm 0.89$ | $31.46 \pm 1.26$ | $30.32 \pm 0.64$ |
| 100Leaves | $72.45 \pm 1.52$ | $81.29 \pm 1.08$ | $93.12 \pm 0.72$ | $84.70 \pm 0.69$ | $93.23 \pm 0.69$ | $93.08 \pm 0.69$ | $92.82 \pm 0.72$ | $\mathbf{94.28 \pm 0.48}$ |
| HW-2view | $71.38 \pm 4.37$ | $76.69 \pm 4.56$ | $67.78 \pm 0.65$ | $81.69 \pm 0.20$ | $62.77 \pm 0.40$ | $66.22 \pm 0.45$ | $63.66 \pm 0.25$ | $\mathbf{83.01 \pm 0.14}$ |
| Cora | $0.55 \pm 0.15$ | $0.60 \pm 0.13$ | $8.32 \pm 0.09$ | $21.12 \pm 1.17$ | $14.29 \pm 0.15$ | $21.40 \pm 0.36$ | $17.68 \pm 0.56$ | $\mathbf{26.62 \pm 0.25}$ |
| Scene15 | $39.02 \pm 0.58$ | $40.20 \pm 0.56$ | $39.53 \pm 0.89$ | $37.48 \pm 0.75$ | $38.93 \pm 0.48$ | $39.63 \pm 0.66$ | $39.15 \pm 0.60$ | $\mathbf{46.22 \pm 0.43}$ |
| CCV | $10.70 \pm 0.25$ | $11.11 \pm 0.33$ | $16.90 \pm 0.40$ | $19.57 \pm 0.41$ | $17.94 \pm 0.52$ | $18.09 \pm 0.32$ | $18.01 \pm 0.30$ | $\mathbf{22.63 \pm 0.25}$ |
| Hdigit | $72.45 \pm 4.90$ | $74.73 \pm 3.21$ | $64.57 \pm 0.46$ | $76.52 \pm 1.32$ | $65.53 \pm 0.56$ | $64.42 \pm 0.13$ | $61.98 \pm 0.10$ | $\mathbf{82.86 \pm 0.02}$ |
| ALOI100 | $71.51 \pm 2.61$ | $80.17 \pm 0.47$ | $84.61 \pm 0.49$ | $78.94 \pm 0.61$ | $84.54 \pm 0.77$ | $83.90 \pm 0.59$ | $84.34 \pm 0.40$ | $\mathbf{87.18 \pm 0.49}$ |
| | | | | **Purity (%)** | | | | |
| Reuters | $42.43 \pm 3.24$ | $44.60 \pm 2.80$ | $51.27 \pm 1.00$ | $\mathbf{53.36 \pm 0.07}$ | $50.33 \pm 0.00$ | $50.65 \pm 0.95$ | $52.29 \pm 1.94$ | $51.17 \pm 2.10$ |
| 100Leaves | $49.24 \pm 2.64$ | $65.11 \pm 1.94$ | $85.54 \pm 1.67$ | $71.64 \pm 1.04$ | $85.33 \pm 1.65$ | $85.42 \pm 1.36$ | $84.78 \pm 1.29$ | $\mathbf{87.26 \pm 0.90}$ |
| HW-2view | $75.27 \pm 5.88$ | $82.33 \pm 5.87$ | $76.06 \pm 0.56$ | $89.92 \pm 0.21$ | $71.98 \pm 0.55$ | $74.51 \pm 0.38$ | $72.29 \pm 0.58$ | $\mathbf{91.86 \pm 0.07}$ |
| Cora | $30.28 \pm 0.10$ | $30.27 \pm 0.13$ | $37.41 \pm 0.06$ | $45.43 \pm 1.22$ | $43.02 \pm 0.20$ | $48.14 \pm 1.33$ | $44.35 \pm 1.30$ | $\mathbf{50.87 \pm 0.18}$ |
| Scene15 | $43.01 \pm 0.83$ | $44.61 \pm 1.00$ | $45.14 \pm 1.46$ | $42.98 \pm 0.94$ | $43.07 \pm 0.86$ | $44.07 \pm 1.24$ | $43.60 \pm 1.06$ | $\mathbf{52.89 \pm 0.38}$ |
| CCV | $19.57 \pm 0.34$ | $19.91 \pm 0.55$ | $23.74 \pm 0.53$ | $26.44 \pm 0.44$ | $24.74 \pm 0.70$ | $25.10 \pm 0.63$ | $25.01 \pm 0.52$ | $\mathbf{31.08 \pm 0.49}$ |
| Hdigit | $76.70 \pm 5.84$ | $79.50 \pm 4.49$ | $73.91 \pm 0.90$ | $82.89 \pm 1.73$ | $74.75 \pm 0.29$ | $73.78 \pm 0.07$ | $71.99 \pm 0.07$ | $\mathbf{92.11 \pm 0.01}$ |
| ALOI100 | $52.32 \pm 3.14$ | $68.22 \pm 1.44$ | $77.25 \pm 0.97$ | $69.54 \pm 1.42$ | $76.82 \pm 1.85$ | $75.65 \pm 1.20$ | $76.50 \pm 1.10$ | $\mathbf{79.96 \pm 1.10}$ |
| | | | | **F-score (%)** | | | | |
| Reuters | $29.70 \pm 2.05$ | $31.38 \pm 1.77$ | $37.08 \pm 0.76$ | $\mathbf{38.04 \pm 0.04}$ | $35.67 \pm 0.00$ | $36.58 \pm 0.69$ | $37.17 \pm 1.03$ | $36.40 \pm 0.99$ |
| 100Leaves | $35.89 \pm 2.63$ | $50.52 \pm 2.54$ | $78.98 \pm 2.23$ | $59.34 \pm 1.63$ | $79.05 \pm 1.99$ | $78.91 \pm 1.79$ | $78.28 \pm 1.20$ | $\mathbf{81.95 \pm 1.29}$ |
| HW-2view | $65.70 \pm 6.26$ | $73.47 \pm 6.73$ | $65.43 \pm 0.85$ | $81.62 \pm 0.33$ | $58.80 \pm 0.70$ | $62.75 \pm 0.52$ | $59.56 \pm 0.59$ | $\mathbf{84.59 \pm 0.13}$ |
| Cora | $19.29 \pm 1.87$ | $18.21 \pm 0.72$ | $20.27 \pm 0.11$ | $28.32 \pm 0.58$ | $24.24 \pm 0.14$ | $28.88 \pm 0.43$ | $26.01 \pm 0.54$ | $\mathbf{33.31 \pm 0.16}$ |
| Scene15 | $28.19 \pm 0.68$ | $29.11 \pm 0.60$ | $29.41 \pm 1.02$ | $27.73 \pm 0.69$ | $28.37 \pm 0.49$ | $29.27 \pm 0.84$ | $28.85 \pm 0.70$ | $\mathbf{35.55 \pm 0.19}$ |
| CCV | $10.11 \pm 0.19$ | $9.87 \pm 0.31$ | $11.70 \pm 0.30$ | $13.85 \pm 0.26$ | $12.36 \pm 0.35$ | $12.50 \pm 0.25$ | $12.46 \pm 0.21$ | $\mathbf{15.70 \pm 0.17}$ |
| Hdigit | $67.22 \pm 7.07$ | $70.61 \pm 5.01$ | $61.58 \pm 0.69$ | $74.11 \pm 2.07$ | $63.00 \pm 0.83$ | $61.38 \pm 0.11$ | $59.02 \pm 0.06$ | $\mathbf{85.20 \pm 0.01}$ |
| ALOI100 | $43.25 \pm 3.71$ | $56.74 \pm 1.17$ | $65.16 \pm 1.42$ | $52.17 \pm 1.44$ | $64.67 \pm 1.55$ | $63.66 \pm 1.63$ | $64.51 \pm 1.26$ | $\mathbf{70.01 \pm 1.51}$ |

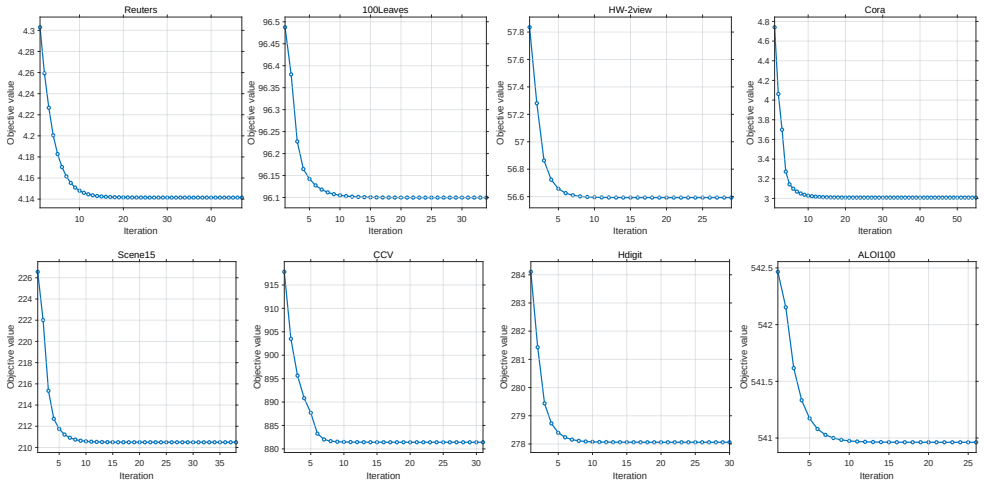



Figure 1: Convergence curves of SMKKM-Sha on eight datasets.

### 4.4 Effect of the Compression Rank

To examine the effect of the compression rank $r$, we vary $r$ on CCV and HW-2Views and report the ACC of SMKKM-Ind and SMKKM-Sha. As shown in Fig. 2, both methods remain stable over a

Table 2: Running time and speedup relative to SMKKM on the benchmark datasets.

| **Datasets** | MKKM | RMKKM | LMKKM | LMTC | CMKC | SMKKM | **SMKKM-Ind** | **SMKKM-Sha** |
|---|---|---|---|---|---|---|---|---|
| | | | | **Running Time (s)** | | | | |
| Reuters | 2.681 | 2.724 | 6.955 | 19.061 | 10.629 | 46.286 | 0.178 | 0.286 |
| 100Leaves | 3.800 | 10.061 | 8.322 | 18.598 | 6.148 | 31.184 | 0.751 | 0.894 |
| HW-2view | 2.873 | 6.723 | 8.558 | 1.379 | 2.417 | 22.353 | 0.193 | 0.252 |
| Cora | 6.936 | 13.735 | 31.140 | 25.753 | 27.011 | 273.499 | 0.584 | 0.891 |
| Scene15 | 38.036 | 44.002 | 72.689 | 4.644 | 8.897 | 212.908 | 1.247 | 1.749 |
| CCV | 22.575 | 88.545 | 142.709 | 8.329 | 2.148 | 115.494 | 1.244 | 0.943 |
| Hdigit | 98.480 | 147.822 | 430.698 | 8.936 | 12.462 | 611.855 | 4.301 | 5.156 |
| ALOI100 | 463.359 | 495.077 | 694.397 | 82.126 | 31.871 | 1581.531 | 13.699 | 16.557 |
| | | | | **Speedup over SMKKM** | | | | |
| Reuters | ×17.26 | ×16.99 | ×6.66 | ×2.43 | ×4.35 | ×1.00 | ×259.66 | ×161.79 |
| 100Leaves | ×8.21 | ×3.10 | ×3.75 | ×1.68 | ×5.07 | ×1.00 | ×41.52 | ×34.87 |
| HW-2view | ×7.78 | ×3.33 | ×2.61 | ×16.21 | ×9.25 | ×1.00 | ×115.58 | ×88.76 |
| Cora | ×39.43 | ×19.91 | ×8.78 | ×10.62 | ×10.13 | ×1.00 | ×468.67 | ×307.07 |
| Scene15 | ×5.60 | ×4.84 | ×2.93 | ×45.85 | ×23.93 | ×1.00 | ×170.79 | ×121.75 |
| CCV | ×5.12 | ×1.30 | ×0.81 | ×13.87 | ×53.76 | ×1.00 | ×92.83 | ×122.43 |
| Hdigit | ×6.21 | ×4.14 | ×1.42 | ×68.47 | ×49.10 | ×1.00 | ×142.25 | ×118.66 |
| ALOI100 | ×3.41 | ×3.19 | ×2.28 | ×19.26 | ×49.62 | ×1.00 | ×115.45 | ×95.52 |
| **Geo. Mean** | ×8.47 | ×4.82 | ×2.85 | ×12.48 | ×16.85 | ×1.00 | ×140.52 | ×112.68 |

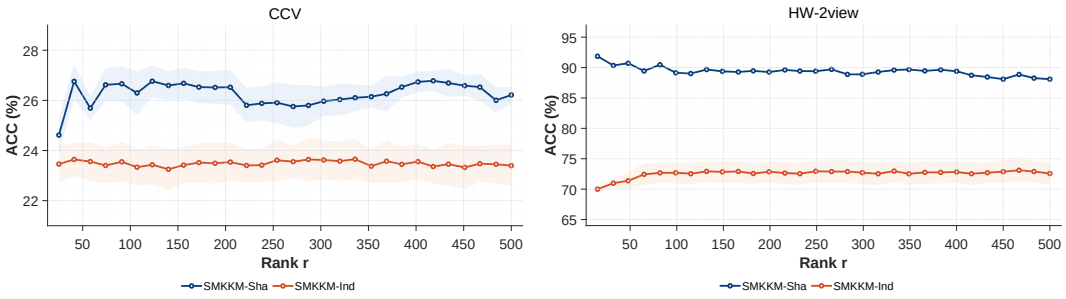



Figure 2: Effect of the compression rank $r$ on the ACC of SMKKM-Ind and SMKKM-Sha on CCV and HW-2Views. The solid line denotes the mean ACC, and the shaded region represents the standard deviation.

wide range of ranks and achieve results close to their best performance even with relatively small $r$. SMKKM-Sha has a higher average ACC than SMKKM-Ind on both datasets, while increasing $r$ does not lead to a steady improvement in performance. These results show that both methods are not sensitive to the choice of $r$, and that a compact representation is enough to preserve the main clustering information while keeping the computational cost low.

## 5 CONCLUSION

We propose a scalable multiple kernel clustering framework based on symmetric randomized compression, with kernel fusion and spectral optimization performed in a compact core space. Theoretical analysis characterizes the compression error and the cross-kernel consistency of the shared subspace. Experiments show that the proposed method maintains or improves clustering performance while substantially accelerating SimpleMKKM. Future work will consider adaptive rank selection and extensions to streaming and incomplete multi-view settings.

## AI USE STATEMENT

In this work, we used generative AI tools solely for language polishing, including improving grammar, clarity, and readability. Generative AI was not used for research idea generation, method design, experiments, code, result analysis, or citation generation. All AI-assisted edits were reviewed and verified by the authors. We take full responsibility for the final content of this work.

## REFERENCES


Massih R Amini, Nicolas Usunier, and Cyril Goutte. Learning from multiple partially observed views-an application to multilingual text categorization. *Advances in neural information processing systems*, 22, 2009.

Arthur Asuncion, David Newman, et al. Uci machine learning repository, 2007.

Liang Du, Peng Zhou, Lei Shi, Hanmo Wang, Mingyu Fan, Wenjian Wang, and Yi-Dong Shen. Robust multiple kernel k-means using l21-norm. In *IJCAI*, volume 15, pp. 3476–3482, 2015.

Liang Du, Yukai Shi, Yan Chen, Feijiang Li, Peng Zhou, Lei Duan, and Yuhua Qian. A dual mixture-of-experts framework for multi-view k-means clustering with view balance and regional sparsity. *Information Fusion*, 125:103449, 2026.

Yu Feng, Weixuan Liang, Xinhang Wan, Jiyuan Liu, Suyuan Liu, Qian Qu, Renxiang Guan, Huiying Xu, and Xinwang Liu. Incremental nyström-based multiple kernel clustering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 16613–16621, 2025.

Jan-Mark Geusebroek, Gertjan J Burghouts, and Arnold WM Smeulders. The amsterdam library of object images. *International Journal of Computer Vision*, 61(1):103–112, 2005.

Mehmet Gönen and Adam A. Margolin. Localized data fusion for kernel $k$-means clustering with application to cancer biology. In *Advances in Neural Information Processing Systems*, volume 27, pp. 1305–1313, 2014.

Nathan Halko, Per-Gunnar Martinsson, and Joel A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53 (2):217–288, 2011. doi: 10.1137/090771806.

Hsin-Chien Huang, Yung-Yu Chuang, and Chu-Song Chen. Multiple kernel fuzzy clustering. *IEEE Transactions on Fuzzy Systems*, 20(1):120–134, 2012. doi: 10.1109/TFUZZ.2011.2170175.

Jintian Ji and Songhe Feng. Anchors crash tensor: Efficient and scalable tensorial multi-view subspace clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(4):2660–2675, 2025.

Yu-Gang Jiang, Guangnan Ye, Shih-Fu Chang, Daniel Ellis, and Alexander C Loui. Consumer video understanding: A benchmark database and an evaluation of human and machine performance. In *Proceedings of the 1st ACM international conference on multimedia retrieval*, pp. 1–8, 2011.

Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, volume 2, pp. 2169–2178. Ieee, 2006.

Weixuan Liang, Xinwang Liu, Yong Liu, Chuan Ma, Yunping Zhao, Zhe Liu, and En Zhu. Consistency of multiple kernel clustering. In *International Conference on Machine Learning*, pp. 20650–20676. PMLR, 2023.

Weixuan Liang, En Zhu, Shengju Yu, Huiying Xu, Xinzhong Zhu, and Xinwang Liu. Scalable multiple kernel clustering: Learning clustering structure from expectation. In *Forty-first International Conference on Machine Learning*, 2024.

Weixuan Liang, Xinwang Liu, Ke Liang, Jiyuan Liu, and En Zhu. Coke: Core kernel for more efficient approximation of kernel weights in multiple kernel clustering. In *Forty-second International Conference on Machine Learning*, 2025.

Jiyuan Liu, Xinwang Liu, Jian Xiong, Qing Liao, Sihang Zhou, Siwei Wang, and Yuexiang Yang. Optimal neighborhood multiple kernel clustering with adaptive local kernels. *IEEE Transactions on Knowledge and Data Engineering*, 34(6):2872–2885, 2020.

Jiyuan Liu, Xinwang Liu, Yuexiang Yang, Qing Liao, and Yuanqing Xia. Contrastive multi-view kernel learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(8):9552–9566, 2023.

Jiyuan Liu, Xinwang Liu, Chuankun Li, Xinhang Wan, Hao Tan, Yi Zhang, Weixuan Liang, Qian Qu, Yu Feng, Renxiang Guan, et al. Large-scale multi-view tensor clustering with implicit linear kernels. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20727–20736. IEEE, 2025a.

Jiyuan Liu, Xinwang Liu, Siqi Wang, Xinhang Wan, Dongsheng Li, Kai Lu, and Kunlun He. Communication-efficient federated multi-view clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025b.

Suyuan Liu, Siwei Wang, Ke Liang, Xueling Zhu, Jinyan Li, Xinwang Liu, and Xiaohong Chen. Propagating cross-view semantics for multi-view clustering: A unified anchor refinement paradigm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

Xinwang Liu. SimpleMKKM: Simple multiple kernel $k$-means. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):5174–5186, 2023. doi: 10.1109/TPAMI.2022.3198638.

Xinwang Liu, Yong Dou, Jianping Yin, Lei Wang, and En Zhu. Multiple kernel $k$-means clustering with matrix-induced regularization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, pp. 1888–1894, 2016. doi: 10.1609/aaai.v30i1.10249.

Xinwang Liu, Xinzhong Zhu, Miaomiao Li, Lei Wang, En Zhu, Tongliang Liu, Marius Kloft, Dinggang Shen, Jianping Yin, and Wen Gao. Multiple kernel $k$ k-means with incomplete kernels. *IEEE transactions on pattern analysis and machine intelligence*, 42(5):1191–1204, 2019.

Charles Mallah, James Cope, James Orwell, et al. Plant leaf classification using probabilistic integration of shape, texture and margin features. *Signal Processing, Pattern Recognition and Applications*, 5(1):45–54, 2013.

Andrew Kachites McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3(2):127–163, 2000.

Feiping Nie, Jingjing Xue, Weizhong Yu, and Xuelong Li. Fast clustering with anchor guidance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(4):1898–1912, 2023.

Zhenwen Ren, Quansen Sun, and Dong Wei. Multiple kernel clustering with kernel k-means coupled graph tensor learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 9411–9418, 2021.

Jun Wang, Chang Tang, Xiao Zheng, Xinwang Liu, Wei Zhang, En Zhu, and Xinzhong Zhu. Fast approximated multiple kernel k-means. *IEEE Transactions on Knowledge and Data Engineering*, 36(11):6171–6180, 2023.

Siwei Wang, Xinwang Liu, Qing Liao, Yi Wen, En Zhu, and Kunlun He. Scalable multi-view graph clustering with cross-view corresponding anchor alignment. *IEEE Transactions on Knowledge and Data Engineering*, 37(5):2932–2945, 2025.

Jie Wen, Jiang Long, Xiaohuan Lu, Chengliang Liu, Xiaozhao Fang, and Yong Xu. Partial multiview incomplete multilabel learning via uncertainty-driven reliable dynamic fusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Yaqiang Yao, Yang Li, Bingbing Jiang, and Huanhuan Chen. Multiple kernel k-means clustering by selecting representative kernels. *IEEE Transactions on Neural Networks and Learning Systems*, 32(11):4983–4996, 2020.

Shi Yu, Léon-Charles Tranchevent, Xinhai Liu, Wolfgang Glänzel, Johan A. K. Suykens, Bart De Moor, and Yves Moreau. Optimized data fusion for kernel $k$-means clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(5):1031–1039, 2012. doi: 10.1109/TPAMI.2011.255.

# A Appendix

### A.1 Theoretical Proofs

#### A.1.1 Single-kernel approximation and Gaussian guarantees

We consider fixed positive semidefinite kernels with eigenvalues $\lambda_{p,1} \geq \cdots \geq \lambda_{p,n} \geq 0$ and set $\lambda_{p,n+1} = 0$. All expectations and probabilities are over the Gaussian sketch. Assume $k \leq r \leq \ell = r + s \leq n$ and $\operatorname{rank}(\mathbf{K}_p) \geq \ell$, so that each sampled basis has $\ell$ columns almost surely. Here $s$ is the effective oversampling after capping the sketch width at $n$. Local compression uses the rank-$r$ Rayleigh–Ritz basis in Definition 2, whereas shared compression uses the full sampled bases $\mathbf{Z}_p$. For rank-deficient samples, range residuals use the actual sampled range; any extra directions supplied by QR can affect the consensus basis.

For any orthonormal $\mathbf{Q}$ and $\mathbf{K} \succeq 0$, both $\mathbf{C} = \mathbf{Q}^\top \mathbf{K} \mathbf{Q}$ and $\widehat{\mathbf{K}} = \mathbf{Q}\mathbf{C}\mathbf{Q}^\top$ are positive semidefinite, with $\operatorname{rank}(\widehat{\mathbf{K}}) = \operatorname{rank}(\mathbf{C}) \leq r$. We distinguish the sampling residual from the final reconstruction error:

$$\varepsilon_p = \|(\mathbf{I} - \mathbf{Z}_p \mathbf{Z}_p^\top)\mathbf{K}_p\|_2, \qquad d_p = \|\mathbf{K}_p - \widehat{\mathbf{K}}_p\|_2. \tag{29}$$

*Proof of Theorem 1.* Write $\mathbf{P}_Z = \mathbf{Z}_p \mathbf{Z}_p^\top$. For any orthogonal projector $\mathbf{P}_Z$ and symmetric $\mathbf{K}_p$,

$$\mathbf{K}_p - \mathbf{P}_Z \mathbf{K}_p \mathbf{P}_Z = (\mathbf{I} - \mathbf{P}_Z)\mathbf{K}_p + \mathbf{P}_Z \mathbf{K}_p (\mathbf{I} - \mathbf{P}_Z), \tag{30}$$

$$\|\mathbf{K}_p - \mathbf{P}_Z \mathbf{K}_p \mathbf{P}_Z\|_2 \leq 2\|(\mathbf{I} - \mathbf{P}_Z)\mathbf{K}_p\|_2 = 2\varepsilon_p. \tag{31}$$

Let $(\mathbf{T}_p)_r$ retain the leading $r$ eigenpairs of $\mathbf{T}_p = \mathbf{Z}_p^\top \mathbf{K}_p \mathbf{Z}_p$. By construction, $\widehat{\mathbf{K}}_p = \mathbf{Z}_p (\mathbf{T}_p)_r \mathbf{Z}_p^\top$. Isometry and eigenvalue interlacing imply

$$\|\mathbf{Z}_p[\mathbf{T}_p - (\mathbf{T}_p)_r]\mathbf{Z}_p^\top\|_2 = \lambda_{r+1}(\mathbf{T}_p) \leq \lambda_{p,r+1}.$$

If the core has at most $r$ eigenvalues, this truncation residual is zero. The triangle inequality proves the stated bound. □

**Corollary 1** (Expected error of Gaussian range finding)**.** *For $s \geq 2$, $\ell = r + s \leq n$, and $h = q + 1$, define*

$$\mathcal{B}_p(r, s, q) = \left[ \left(1 + \sqrt{\frac{r}{s-1}}\right) \lambda_{p,r+1}^h + \frac{e\sqrt{r+s}}{s} \left( \sum_{j>r} \lambda_{p,j}^{2h} \right)^{1/2} \right]^{1/h}. \tag{32}$$

*Then*

$$\mathbb{E}\varepsilon_p \leq \mathcal{B}_p(r, s, q), \qquad \mathbb{E}d_p \leq 2\mathcal{B}_p(r, s, q) + \lambda_{p,r+1}. \tag{33}$$

For $q = 0$, the first inequality follows from Halko et al. (2011, Theorem 10.6); the argument below covers $q > 0$. The second follows from Theorem 1, whose deterministic bound also holds for $s = 0$ or $s = 1$.

**Power inequality for the expected range error.** Let $h = q + 1 \geq 1$ and $\mathbf{P}_Z = \mathbf{Z}_p \mathbf{Z}_p^\top$. For a unit vector $\mathbf{x}$ in $\operatorname{range}(\mathbf{I} - \mathbf{P}_Z)$, the scalar power-mean inequality in an eigenbasis of $\mathbf{K}_p$ yields $\|\mathbf{K}_p \mathbf{x}\|_2 \leq \|\mathbf{K}_p^h \mathbf{x}\|_2^{1/h}$. Taking suprema over such vectors, using symmetry and then Jensen's inequality, gives

$$\mathbb{E}\varepsilon_p \leq \mathbb{E}\|(\mathbf{I} - \mathbf{P}_Z)\mathbf{K}_p^h\|_2^{1/h} \leq \left(\mathbb{E}\|(\mathbf{I} - \mathbf{P}_Z)\mathbf{K}_p^h\|_2\right)^{1/h}.$$

Applying the Gaussian range-finder estimate to $\mathbf{K}_p^h$ proves Eq. (32) for the implemented exponent $h = q + 1$.

**Simultaneous probability guarantee.** For $s \geq 2$ and $0 < \delta < 1$, choose $\delta_p > 0$ with $\sum_p \delta_p \leq \delta$. Markov's inequality and a union bound give, with probability at least $1 - \delta$, simultaneously for all kernels,

$$\varepsilon_p \leq \frac{\mathcal{B}_p(r, s, q)}{\delta_p}, \qquad d_p \leq D_p(\delta_p) := \frac{2\mathcal{B}_p(r, s, q)}{\delta_p} + \lambda_{p,r+1}. \tag{34}$$

One may take $\delta_p = \delta/m$. The union bound allows a single Gaussian test matrix to be reused across kernels.

The bound depends on the next omitted eigenvalue and the spectral tail. If $\operatorname{rank}(\mathbf{K}_p) \leq r$, the Gaussian sketch captures the kernel range almost surely and reconstruction is exact.

### A.1.2 Uniform Error over the Base-Kernel Collection

Define the largest reconstruction error among the $m$ base kernels as

$$d_{\max} := \max_{1\le p\le m} \|\mathbf{K}_p - \widehat{\mathbf{K}}_p\|_2. \tag{35}$$

Applying Theorem 1 to every base kernel gives

$$d_{\max} \le \max_{1\le p\le m} \left(2\varepsilon_p + \lambda_{p,r+1}\right). \tag{36}$$

Thus, the approximation quality of the complete kernel collection is controlled by the base kernel with the largest sampled-range residual or spectral tail. On the simultaneous event in Eq. (34), the same statement becomes $d_{\max} \le \max_p D_p(\delta_p)$ with probability at least $1-\delta$.

### A.1.3 Proof of Proposition 1

*Proof.* Using the relationship between the Frobenius norm and the trace, we obtain

$$\begin{aligned}\sum_{p=1}^{m} \left\|\mathbf{Q}^\top \mathbf{Z}_p\right\|_F^2 &= \sum_{p=1}^{m} \mathrm{Tr}\big(\mathbf{Q}^\top \mathbf{Z}_p \mathbf{Z}_p^\top \mathbf{Q}\big) \\ &= \mathrm{Tr}\left[\mathbf{Q}^\top \left(\sum_{p=1}^{m} \mathbf{Z}_p \mathbf{Z}_p^\top\right) \mathbf{Q}\right].\end{aligned} \tag{37}$$

By the definition of the concatenated basis,

$$\mathbf{Z}_{\mathrm{cat}} \mathbf{Z}_{\mathrm{cat}}^\top = \sum_{p=1}^{m} \mathbf{Z}_p \mathbf{Z}_p^\top. \tag{38}$$

Therefore, Eq. (11) is equivalent to

$$\max_{\mathbf{Q}^\top \mathbf{Q} = \mathbf{I}_r} \mathrm{Tr}\big(\mathbf{Q}^\top \mathbf{Z}_{\mathrm{cat}} \mathbf{Z}_{\mathrm{cat}}^\top \mathbf{Q}\big). \tag{39}$$

By the Rayleigh–Ritz theorem, an optimizer is formed by the $r$ leading eigenvectors of $\mathbf{Z}_{\mathrm{cat}} \mathbf{Z}_{\mathrm{cat}}^\top$. Moreover, the singular value decomposition

$$\mathbf{Z}_{\mathrm{cat}} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^\top$$

implies

$$\mathbf{Z}_{\mathrm{cat}} \mathbf{Z}_{\mathrm{cat}}^\top = \mathbf{U} \mathbf{\Sigma}^2 \mathbf{U}^\top.$$

Hence, its $r$ leading eigenvectors are precisely the $r$ leading left singular vectors of $\mathbf{Z}_{\mathrm{cat}}$. It follows that

$$\mathbf{Q}^\star = \mathbf{U}_{(:,1:r)}.$$

□

### A.1.4 Shared fusion consistency and subspace recovery

Proposition 2 follows directly from linearity:

$$\sum_{p=1}^{m} \gamma_p^2 \mathbf{C}_p^{(\mathrm{sha})} = \sum_{p=1}^{m} \gamma_p^2 \mathbf{Q}^\top \mathbf{K}_p \mathbf{Q} = \mathbf{Q}^\top \mathbf{K}_{\boldsymbol{\gamma}} \mathbf{Q}.$$

**Proof of Proposition 3.**

*Proof.* Write $\mathbf{U}_k = \mathbf{U}_{\boldsymbol{\gamma},k}$ and $\mathbf{K}_{\boldsymbol{\gamma}} \mathbf{U}_k = \mathbf{U}_k \mathbf{\Lambda}_k$. Under Eq. (15), $\mathbf{U}_k = \mathbf{Q}\mathbf{H}_k$ with $\mathbf{H}_k = \mathbf{Q}^\top \mathbf{U}_k$ and $\mathbf{H}_k^\top \mathbf{H}_k = \mathbf{I}_k$. Therefore,

$$\mathbf{C}_{\boldsymbol{\gamma}}^{(\mathrm{sha})} \mathbf{H}_k = \mathbf{Q}^\top \mathbf{K}_{\boldsymbol{\gamma}} \mathbf{Q} \mathbf{H}_k = \mathbf{Q}^\top \mathbf{K}_{\boldsymbol{\gamma}} \mathbf{U}_k = \mathbf{H}_k \mathbf{\Lambda}_k.$$

Thus the compressed core contains these $k$ leading eigenvalues and their orthonormal eigenvectors. By the Rayleigh–Ritz principle, an orthogonal compression cannot have a larger top-$k$ eigenvalue sum than $\mathbf{K}_{\boldsymbol{\gamma}}$; hence $\mathbf{H}_k$ attains the compressed top-$k$ optimum. Finally, $\mathbf{Q}\mathbf{H}_k = \mathbf{U}_k$ gives Eq. (16). A positive eigengap makes the leading invariant subspace unique up to rotation. □

### A.2 ALGORITHM FRAMEWORK

Algorithms 1 and 2 summarize the two compression and recovery schemes. Both use the reduced-gradient weight updates described in the Optimization subsection.

**Algorithm 1:** SMKKM-Ind: independent kernel compression

**Input:** Base kernels $\{\mathbf{K}_p\}_{p=1}^m$, number of clusters $k$, rank $r$, oversampling size $s$, and power parameter $q$.
**Output:** Cluster assignments and learned weights $\boldsymbol{\gamma}^{(\mathrm{ind})}$.
Set $\ell = \min\{n, r+s\}$ and draw $\boldsymbol{\Omega} \in \mathbb{R}^{n\times\ell}$ with Gaussian entries.
For each $p$, compute $\mathbf{Z}_p = \mathrm{orth}(\mathbf{K}_p^{q+1}\boldsymbol{\Omega})$, $\mathbf{T}_p = \mathbf{Z}_p^\top \mathbf{K}_p \mathbf{Z}_p$, $\mathbf{Q}_p = \mathbf{Z}_p \mathbf{V}_{p,r}$, and $\mathbf{C}_p^{(\mathrm{ind})} = \mathbf{Q}_p^\top \mathbf{K}_p \mathbf{Q}_p$, where $\mathbf{V}_{p,r}$ contains the leading $r$ eigenvectors of $\mathbf{T}_p$.
Initialize $\gamma_p = 1/m$ for all $p$.
Repeat until convergence: form $\mathbf{C}_{\boldsymbol{\gamma}}^{(\mathrm{ind})} = \sum_p \gamma_p^2 \mathbf{C}_p^{(\mathrm{ind})}$, compute its leading $k$ eigenvectors, and update $\boldsymbol{\gamma}$ by feasible reduced-gradient descent with line search.
Factor $\mathbf{C}_p^{(\mathrm{ind})} = \mathbf{R}_p \mathbf{R}_p^\top$ for each $p$ and form $\mathbf{L}_{\boldsymbol{\gamma}} = [\gamma_1 \mathbf{Q}_1 \mathbf{R}_1, \ldots, \gamma_m \mathbf{Q}_m \mathbf{R}_m]$.
Compute the leading $k$ left singular vectors $\mathbf{F}^{(\mathrm{ind})}$ of $\mathbf{L}_{\boldsymbol{\gamma}}$.
Apply $k$-means to the rows of $\mathbf{F}^{(\mathrm{ind})}$.

**Algorithm 2:** SMKKM-Sha: shared-subspace compression

**Input:** Base kernels $\{\mathbf{K}_p\}_{p=1}^m$, number of clusters $k$, rank $r$, oversampling size $s$, and power parameter $q$.
**Output:** Cluster assignments and learned weights $\boldsymbol{\gamma}^{(\mathrm{sha})}$.
Set $\ell = \min\{n, r+s\}$, draw $\boldsymbol{\Omega} \in \mathbb{R}^{n\times\ell}$ with Gaussian entries, and compute $\mathbf{Z}_p = \mathrm{orth}(\mathbf{K}_p^{q+1}\boldsymbol{\Omega})$ for each $p$.
Compute the SVD of $\mathbf{Z}_{\mathrm{cat}} = [\mathbf{Z}_1, \ldots, \mathbf{Z}_m]$ and take its leading $r$ left singular vectors as $\mathbf{Q}$.
For each $p$, form the shared core $\mathbf{C}_p^{(\mathrm{sha})} = \mathbf{Q}^\top \mathbf{K}_p \mathbf{Q}$.
Initialize $\gamma_p = 1/m$ for all $p$.
Repeat until convergence: form $\mathbf{C}_{\boldsymbol{\gamma}}^{(\mathrm{sha})} = \sum_p \gamma_p^2 \mathbf{C}_p^{(\mathrm{sha})}$, compute its leading $k$ eigenvectors, and update $\boldsymbol{\gamma}$ by feasible reduced-gradient descent with line search.
With the converged weights, compute the leading $k$ eigenvectors $\mathbf{H}^{(\mathrm{sha})}$ of $\mathbf{C}_{\boldsymbol{\gamma}}^{(\mathrm{sha})}$ and set $\mathbf{F}^{(\mathrm{sha})} = \mathbf{Q}\mathbf{H}^{(\mathrm{sha})}$.
Apply $k$-means to the rows of $\mathbf{F}^{(\mathrm{sha})}$.

### A.3 TIME COMPLEXITY ANALYSIS

Let $n, m, k, r, \ell$, and $q$ have their definitions above, and let $E$ be the total number of core objective evaluations, including line-search trials. With dense, precomputed kernels, sampling costs $O(m(q+1)n^2\ell + mn\ell^2)$; each subsequent core evaluation costs $O(mr^2 + r^3)$.

**SMKKM-Ind.** Per-kernel projection and rank-$r$ truncation cost $O(mn^2\ell + mn^2 r + m\ell^3)$. The factorized recovery in Eq. (26) takes an exact thin SVD of the $n \times mr$ joint factor, giving

$$\mathcal{T}_{\mathrm{Ind}} = O\big(m(q+2)n^2\ell + mn^2 r + mn\ell^2 + m\ell^3 + E(mr^2 + r^3) + nmr\min\{n, mr\}\big). \tag{40}$$

**SMKKM-Sha.** The consensus SVD of the $n \times m\ell$ concatenated basis costs $O(nm\ell \min\{n, m\ell\})$; shared-core construction costs $O(mn^2 r)$, and final lifting costs $O(nrk)$. Hence,

$$\mathcal{T}_{\mathrm{Sha}} = O\big(m(q+1)n^2\ell + mn\ell^2 + nm\ell\min\{n, m\ell\} + mn^2 r + E(mr^2 + r^3) + nrk\big). \tag{41}$$

Ind thus pays for the final factor SVD, whereas Sha pays for the consensus SVD and directly lifts the core embedding. A Lloyd iteration on either $k$-dimensional embedding adds $O(nk^2)$ per restart. Although the overall complexity still contains an $O(n^2)$ term, this computation is performed only once during initialization. Subsequent spectral decomposition and kernel-weight optimization are carried out in the compact core space, avoiding repeated operations on large kernel matrices. This computational design accounts for the substantial speedups observed in Fig. 3.

To further examine scalability with respect to the number of samples, we gradually increase the sample size on the eight benchmark datasets and record the corresponding running time. As shown in Fig. 4, the running time exhibits an approximately linear growth trend as the sample size increases. Even on larger datasets such as Hdigit and ALOI100, the runtime remains moderate when using all samples, demonstrating the practical scalability of the proposed method.

## A.4 Experimental Protocol

### A.4.1 Datasets

**Reuters:** Derived from the Reuters RCV1/RCV2 corpus, Reuters contains news articles in six categories and five languages: English, French, German, Spanish, and Italian. The languages provide five views of the articles.

**100Leaves:** Its 1,600 leaf images represent 100 plant species, with 16 images per species. Shape descriptors, fine-scale margin features, and texture histograms form the three views.

Table 3: Statistics of the eight datasets.

| Dataset | #Samples | #kernel | #clusters |
|---|---|---|---|
| Reuters-1200 | 1200 | 20 | 6 |
| 100Leaves | 1600 | 12 | 100 |
| HW-2view | 2000 | 8 | 10 |
| Cora | 2708 | 16 | 7 |
| Scene15 | 4485 | 12 | 15 |
| CCV | 6773 | 3 | 20 |
| Hdigit | 10000 | 8 | 10 |
| ALOI-100 | 10800 | 16 | 100 |

**HW-2view:** This handwritten digit dataset contains 2,000 samples from ten classes (0–9). Each sample has two feature representations of dimensions 784 and 256.

**Cora:** Its 2,708 scientific papers span seven research categories. Paper text and citation links provide the two views.

**Scene15:** The 4,485 images cover 15 indoor and outdoor scene categories, including streets, forests, offices, and kitchens. Different visual descriptors provide the views.

**CCV:** Columbia Consumer Video contains 9,317 YouTube videos in 20 semantic categories. Visual, spatio-temporal, and audio descriptors, including SIFT, STIP, and MFCC, form its views.

**Hdigit:** This dataset contains 10,000 handwritten digit samples from ten classes. Its two views are derived from MNIST and USPS and have dimensions 784 and 256, respectively.

**ALOI100:** This 100-category subset of the Amsterdam Library of Object Images shows objects under different viewpoints and lighting conditions. RGB, HSV, color similarity, and Haralick texture descriptors provide its views.

### A.4.2 Experimental Details

For a dataset with $k$ classes, both methods set the target rank to $r = k + 5$, the Gaussian sketch size to $\ell = \min\{n, r + 10\}$, and the power parameter to $q = 2$. For SMKKM-Ind and SMKKM-Sha, we report the mean and standard deviation of ACC, NMI, Purity, and F-score. To reduce the effect of random initialization in $k$-means, all methods are independently run 20 times, and the average performance with standard deviation is reported. The running time includes randomized sketching, core matrix construction, kernel-weight optimization, and recovery of the sample-level spectral representation.

## A.5 Additional Results

### A.5.1 Effect of the Multi-kernel Consensus Subspace

To verify whether the performance improvement comes from the proposed multi-kernel consensus subspace, we only change the construction of the shared subspace $Q$, while keeping the kernel preprocessing, randomized low-rank compression, model optimization, and clustering evaluation

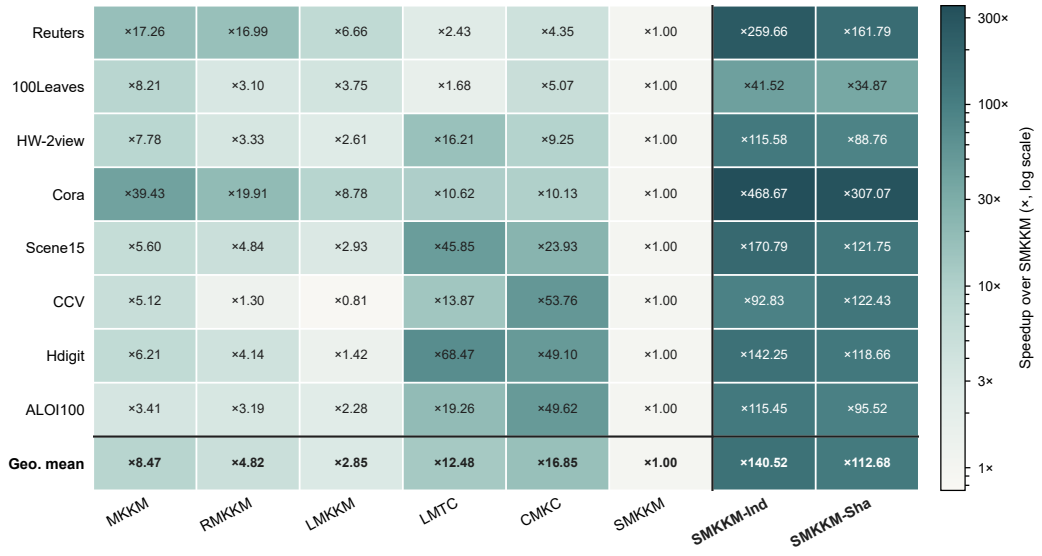



Figure 3: Speedup over SMKKM across the benchmark datasets. Cell colors are shown on a logarithmic scale, while the annotations report the corresponding speedup factors. Gray denotes an unavailable result. The bottom row gives the geometric mean over the datasets with available timings for each method, and the two proposed methods are outlined.

unchanged. For each dataset, we first randomly select three kernels and construct three single-kernel subspaces, one from each kernel. We then form three pairwise kernel combinations to construct three two-kernel consensus subspaces. Finally, all kernels in the dataset are used to construct the full multi-kernel consensus subspace. Table 4 reports the mean and standard deviation of ACC and NMI under different constructions of $Q$.

Overall, the clustering performance improves steadily as more kernel information is used to construct the shared subspace $Q$. Across the eight datasets, the three single-kernel subspace settings achieve average ACC and NMI values of 38.91% and 34.42%, respectively. When two kernels are used to construct $Q$, the average ACC and NMI increase to 52.15% and 48.80%, corresponding to gains of 13.24% and 14.38% percentage points over the single-kernel settings. When all kernels are used to construct the multi-kernel consensus subspace, the average ACC and NMI further increase to 65.14% and 59.14%, with additional gains of 12.99% and 10.34% over the two-kernel settings. These results suggest that different kernels often capture different structural properties of the data. A consensus subspace constructed from all kernels can integrate their complementary information more fully, whereas a single kernel or a specific pair of kernels may preserve only part of the underlying data structure.

### A.5.2 Additional Clustering Metrics

We further report the Precision, Recall, and ARI results of the proposed methods on the eight datasets in Table 5. The proposed methods achieve strong performance on most datasets and evaluation metrics, further demonstrating their effectiveness and stability under different clustering measures.

### A.5.3 Rank Sensitivity of SMKKM-Ind

We further examine the effect of the compression rank $r$ on clustering performance in Fig. 5. As shown in the figure, the clustering performance remains stable over the tested range of $r$ and does not consistently improve as $r$ increases. This indicates that the proposed method is not sensitive to the choice of $r$, and a relatively small rank is sufficient to achieve good clustering performance while reducing computational cost.

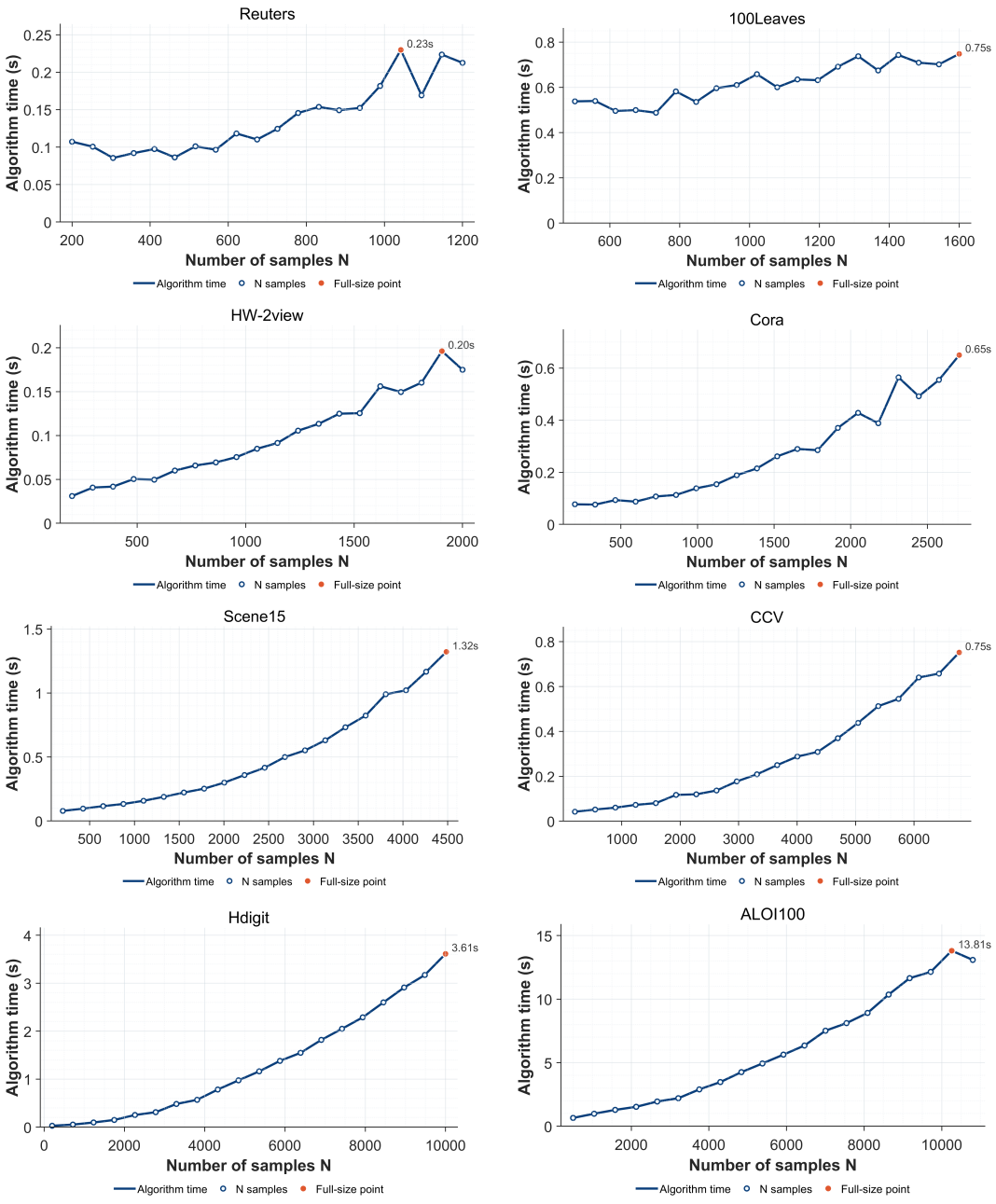



Figure 4: Running time with increasing numbers of samples on the eight benchmark datasets. The highlighted point denotes the result obtained using the full dataset.

Table 4: Clustering performance (ACC and NMI, %) obtained using different subspaces to construct $Q$. Dataset results are reported as mean ± standard deviation; average rows separately average the means and standard deviations across eight datasets. The highest mean in each row is shown in bold.

| | Single-kernel subspace | | | Two-kernel consensus subspace | | | All-kernel consensus |
|---|---|---|---|---|---|---|---|
| Dataset | $Q_{\{k_1\}}$ | $Q_{\{k_2\}}$ | $Q_{\{k_3\}}$ | $Q_{\{k_1,k_2\}}$ | $Q_{\{k_1,k_3\}}$ | $Q_{\{k_2,k_3\}}$ | $Q_{\text{all}}$ |
| **ACC (%)** | | | | | | | |
| Reuters-1200 | $38.94 \pm 0.86$ | $40.82 \pm 1.23$ | $39.26 \pm 0.76$ | $50.57 \pm 0.86$ | $47.34 \pm 0.82$ | $\mathbf{51.50 \pm 1.94}$ | $49.40 \pm 1.18$ |
| 100Leaves | $54.59 \pm 0.86$ | $59.62 \pm 1.22$ | $58.41 \pm 1.37$ | $79.68 \pm 1.43$ | $78.92 \pm 2.21$ | $59.37 \pm 1.67$ | $\mathbf{84.50 \pm 1.10}$ |
| HW | $43.15 \pm 1.60$ | $44.84 \pm 1.95$ | $42.92 \pm 2.10$ | $62.12 \pm 0.26$ | $64.37 \pm 0.31$ | $61.16 \pm 2.67$ | $\mathbf{91.86 \pm 0.07}$ |
| Cora | $28.82 \pm 0.28$ | $40.99 \pm 0.52$ | $25.27 \pm 0.31$ | $49.16 \pm 0.17$ | $33.23 \pm 1.20$ | $46.20 \pm 0.51$ | $\mathbf{49.21 \pm 0.17}$ |
| Scene-15 | $20.65 \pm 0.84$ | $33.98 \pm 0.56$ | $24.70 \pm 0.45$ | $38.45 \pm 1.42$ | $37.63 \pm 1.16$ | $45.33 \pm 0.99$ | $\mathbf{49.00 \pm 0.66}$ |
| CCV | $16.33 \pm 0.30$ | $20.62 \pm 0.55$ | $16.61 \pm 0.24$ | $26.68 \pm 1.18$ | $22.65 \pm 0.53$ | $25.39 \pm 0.58$ | $\mathbf{27.38 \pm 0.56}$ |
| Hdigit | $48.77 \pm 0.13$ | $48.61 \pm 0.66$ | $53.42 \pm 1.83$ | $65.00 \pm 2.49$ | $65.79 \pm 0.01$ | $62.59 \pm 0.02$ | $\mathbf{92.11 \pm 0.01}$ |
| ALOI-100 | $23.31 \pm 0.69$ | $60.60 \pm 1.25$ | $48.68 \pm 1.10$ | $63.25 \pm 1.41$ | $51.94 \pm 1.09$ | $63.28 \pm 1.18$ | $\mathbf{77.69 \pm 1.42}$ |
| **Average** | $34.32 \pm 0.70$ | $43.76 \pm 0.99$ | $38.66 \pm 1.02$ | $54.36 \pm 1.15$ | $50.23 \pm 0.92$ | $51.85 \pm 1.19$ | $\mathbf{65.14 \pm 0.65}$ |
| **NMI (%)** | | | | | | | |
| Reuters-1200 | $20.21 \pm 0.79$ | $18.43 \pm 0.84$ | $16.09 \pm 0.88$ | $30.19 \pm 0.78$ | $29.83 \pm 0.98$ | $27.42 \pm 2.06$ | $\mathbf{30.32 \pm 0.64}$ |
| 100Leaves | $75.86 \pm 0.44$ | $78.66 \pm 0.43$ | $78.20 \pm 0.54$ | $91.06 \pm 0.58$ | $90.74 \pm 0.77$ | $78.60 \pm 0.67$ | $\mathbf{94.28 \pm 0.48}$ |
| HW | $33.72 \pm 0.77$ | $35.20 \pm 1.02$ | $31.06 \pm 1.01$ | $47.90 \pm 0.28$ | $48.48 \pm 0.21$ | $48.06 \pm 1.02$ | $\mathbf{83.01 \pm 0.14}$ |
| Cora | $8.35 \pm 0.27$ | $17.25 \pm 0.34$ | $5.69 \pm 0.16$ | $\mathbf{27.38 \pm 0.12}$ | $12.55 \pm 0.46$ | $26.21 \pm 0.24$ | $26.62 \pm 0.25$ |
| Scene-15 | $13.97 \pm 0.40$ | $32.17 \pm 0.59$ | $17.25 \pm 0.41$ | $39.63 \pm 0.98$ | $32.04 \pm 0.80$ | $38.87 \pm 0.78$ | $\mathbf{46.22 \pm 0.43}$ |
| CCV | $12.26 \pm 0.29$ | $16.86 \pm 0.31$ | $10.94 \pm 0.14$ | $22.51 \pm 0.52$ | $17.86 \pm 0.24$ | $20.54 \pm 0.29$ | $\mathbf{22.63 \pm 0.25}$ |
| Hdigit | $39.96 \pm 0.08$ | $38.44 \pm 0.49$ | $43.72 \pm 0.53$ | $50.82 \pm 0.31$ | $84.42 \pm 0.02$ | $\mathbf{84.65 \pm 0.05}$ | $82.86 \pm 0.02$ |
| ALOI-100 | $40.45 \pm 0.42$ | $74.77 \pm 0.39$ | $66.69 \pm 0.41$ | $76.32 \pm 0.49$ | $68.50 \pm 0.48$ | $76.49 \pm 0.43$ | $\mathbf{87.18 \pm 0.49}$ |
| **Average** | $30.60 \pm 0.43$ | $38.97 \pm 0.55$ | $33.70 \pm 0.51$ | $48.23 \pm 0.51$ | $48.05 \pm 0.49$ | $50.11 \pm 0.69$ | $\mathbf{59.14 \pm 0.34}$ |

Table 5: Clustering performance (%) on the benchmark multi-view datasets. The best and second-best results are highlighted in bold and underlined, respectively.

| **Datasets** | MKKM | RMKKM | LMKKM | LMTC | CMKC | SMKKM | **SMKKM-Ind** | **SMKKM-Sha** |
|---|---|---|---|---|---|---|---|---|
| **Precision (%)** | | | | | | | | |
| Reuters | $26.21 \pm 2.29$ | $27.20 \pm 2.28$ | $\mathbf{36.80 \pm 0.70}$ | $31.95 \pm 0.04$ | $35.03 \pm 0.00$ | $36.31 \pm 0.64$ | $\underline{36.52 \pm 0.71}$ | $33.69 \pm 2.63$ |
| 100Leaves | $28.27 \pm 2.86$ | $44.34 \pm 2.87$ | $\underline{74.56 \pm 2.77}$ | $55.72 \pm 1.84$ | $74.17 \pm 2.50$ | $74.34 \pm 1.92$ | $73.70 \pm 1.58$ | $\mathbf{76.80 \pm 1.46}$ |
| HW-2view | $63.30 \pm 6.18$ | $71.74 \pm 7.36$ | $65.19 \pm 0.83$ | $\underline{81.44 \pm 0.32}$ | $58.25 \pm 0.78$ | $62.37 \pm 0.63$ | $59.08 \pm 0.63$ | $\mathbf{84.53 \pm 0.13}$ |
| Cora | $18.00 \pm 0.19$ | $17.82 \pm 0.13$ | $21.54 \pm 0.09$ | $30.20 \pm 0.29$ | $26.26 \pm 0.14$ | $\underline{31.87 \pm 0.37}$ | $28.79 \pm 0.63$ | $\mathbf{37.16 \pm 0.13}$ |
| Scene15 | $27.36 \pm 0.64$ | $28.55 \pm 0.49$ | $\underline{29.50 \pm 1.02}$ | $27.40 \pm 0.71$ | $28.49 \pm 0.54$ | $29.36 \pm 0.80$ | $28.98 \pm 0.69$ | $\mathbf{35.51 \pm 0.21}$ |
| CCV | $9.44 \pm 0.27$ | $9.83 \pm 0.24$ | $12.39 \pm 0.33$ | $\underline{14.38 \pm 0.33}$ | $13.00 \pm 0.36$ | $13.22 \pm 0.24$ | $13.15 \pm 0.22$ | $\mathbf{16.34 \pm 0.17}$ |
| Hdigit | $65.01 \pm 7.51$ | $68.86 \pm 5.36$ | $61.23 \pm 0.67$ | $\underline{71.01 \pm 2.59}$ | $62.32 \pm 0.90$ | $60.87 \pm 0.10$ | $58.48 \pm 0.10$ | $\mathbf{85.13 \pm 0.02}$ |
| ALOI100 | $34.31 \pm 3.42$ | $51.46 \pm 1.46$ | $\underline{61.10 \pm 2.06}$ | $45.65 \pm 1.76$ | $60.14 \pm 2.00$ | $59.55 \pm 2.16$ | $60.48 \pm 1.91$ | $\mathbf{65.25 \pm 2.26}$ |
| **Recall (%)** | | | | | | | | |
| Reuters | $34.36 \pm 2.21$ | $37.30 \pm 2.53$ | $37.36 \pm 0.82$ | $\mathbf{46.98 \pm 0.07}$ | $36.33 \pm 0.00$ | $36.87 \pm 0.75$ | $37.85 \pm 1.40$ | $\underline{39.88 \pm 1.86}$ |
| 100Leaves | $49.33 \pm 1.72$ | $58.74 \pm 2.13$ | $83.97 \pm 1.68$ | $63.47 \pm 1.59$ | $\underline{84.65 \pm 1.53}$ | $84.09 \pm 1.96$ | $83.48 \pm 1.09$ | $\mathbf{87.84 \pm 1.34}$ |
| HW-2view | $68.34 \pm 6.61$ | $75.36 \pm 6.35$ | $65.68 \pm 0.88$ | $\underline{81.80 \pm 0.34}$ | $59.36 \pm 0.62$ | $63.13 \pm 0.45$ | $60.05 \pm 0.57$ | $\mathbf{84.66 \pm 0.13}$ |
| Cora | $21.19 \pm 4.32$ | $18.68 \pm 1.49$ | $19.14 \pm 0.14$ | $\underline{26.67 \pm 0.84}$ | $22.51 \pm 0.15$ | $26.40 \pm 0.47$ | $23.71 \pm 0.47$ | $\mathbf{30.18 \pm 0.18}$ |
| Scene15 | $29.08 \pm 0.88$ | $\underline{29.71 \pm 0.88}$ | $29.31 \pm 1.02$ | $28.07 \pm 0.68$ | $28.24 \pm 0.44$ | $29.18 \pm 0.91$ | $28.71 \pm 0.73$ | $\mathbf{35.60 \pm 0.24}$ |
| CCV | $10.90 \pm 0.53$ | $9.91 \pm 0.42$ | $11.08 \pm 0.29$ | $\underline{13.35 \pm 0.24}$ | $11.79 \pm 0.36$ | $11.86 \pm 0.27$ | $11.84 \pm 0.22$ | $\mathbf{15.12 \pm 0.18}$ |
| Hdigit | $69.67 \pm 6.86$ | $72.50 \pm 5.00$ | $61.93 \pm 0.72$ | $\underline{77.51 \pm 1.79}$ | $63.69 \pm 0.75$ | $61.89 \pm 0.12$ | $59.57 \pm 0.10$ | $\mathbf{85.27 \pm 0.02}$ |
| ALOI100 | $58.67 \pm 4.50$ | $63.24 \pm 1.00$ | $69.82 \pm 0.92$ | $60.87 \pm 1.00$ | $\underline{69.96 \pm 1.48}$ | $68.41 \pm 1.18$ | $69.14 \pm 0.86$ | $\mathbf{75.56 \pm 0.90}$ |
| **ARI (%)** | | | | | | | | |
| Reuters | $13.36 \pm 2.89$ | $15.06 \pm 2.60$ | $\mathbf{24.44 \pm 0.89}$ | $22.78 \pm 0.05$ | $22.59 \pm 0.00$ | $23.85 \pm 0.81$ | $\underline{24.41 \pm 1.11}$ | $22.42 \pm 2.02$ |
| 100Leaves | $35.11 \pm 2.68$ | $49.98 \pm 2.57$ | $78.77 \pm 2.25$ | $58.92 \pm 1.65$ | $\underline{78.84 \pm 2.01}$ | $78.69 \pm 1.80$ | $78.06 \pm 1.21$ | $\mathbf{81.77 \pm 1.30}$ |
| HW-2view | $61.74 \pm 6.98$ | $70.45 \pm 7.51$ | $61.59 \pm 0.94$ | $\underline{79.58 \pm 0.37}$ | $54.20 \pm 0.78$ | $58.60 \pm 0.59$ | $55.05 \pm 0.66$ | $\mathbf{82.89 \pm 0.14}$ |
| Cora | $0.11 \pm 0.26$ | $-0.13 \pm 0.17$ | $4.09 \pm 0.11$ | $13.84 \pm 0.48$ | $9.22 \pm 0.15$ | $\underline{15.08 \pm 0.46}$ | $11.71 \pm 0.67$ | $\mathbf{20.54 \pm 0.17}$ |
| Scene15 | $22.69 \pm 0.71$ | $23.74 \pm 0.61$ | $\underline{24.18 \pm 1.09}$ | $22.30 \pm 0.74$ | $23.07 \pm 0.53$ | $24.04 \pm 0.89$ | $23.59 \pm 0.75$ | $\mathbf{30.76 \pm 0.21}$ |
| CCV | $4.22 \pm 0.21$ | $4.36 \pm 0.28$ | $6.65 \pm 0.32$ | $\underline{8.81 \pm 0.30}$ | $7.31 \pm 0.37$ | $7.49 \pm 0.26$ | $7.43 \pm 0.22$ | $\mathbf{10.79 \pm 0.18}$ |
| Hdigit | $63.43 \pm 7.93$ | $67.25 \pm 5.59$ | $57.28 \pm 0.77$ | $\underline{71.09 \pm 2.32}$ | $58.84 \pm 0.92$ | $57.05 \pm 0.12$ | $54.43 \pm 0.11$ | $\mathbf{83.56 \pm 0.02}$ |
| ALOI100 | $42.53 \pm 3.77$ | $56.26 \pm 1.18$ | $\underline{64.79 \pm 1.44}$ | $51.62 \pm 1.46$ | $64.29 \pm 1.57$ | $63.27 \pm 1.65$ | $64.13 \pm 1.27$ | $\mathbf{69.69 \pm 1.53}$ |

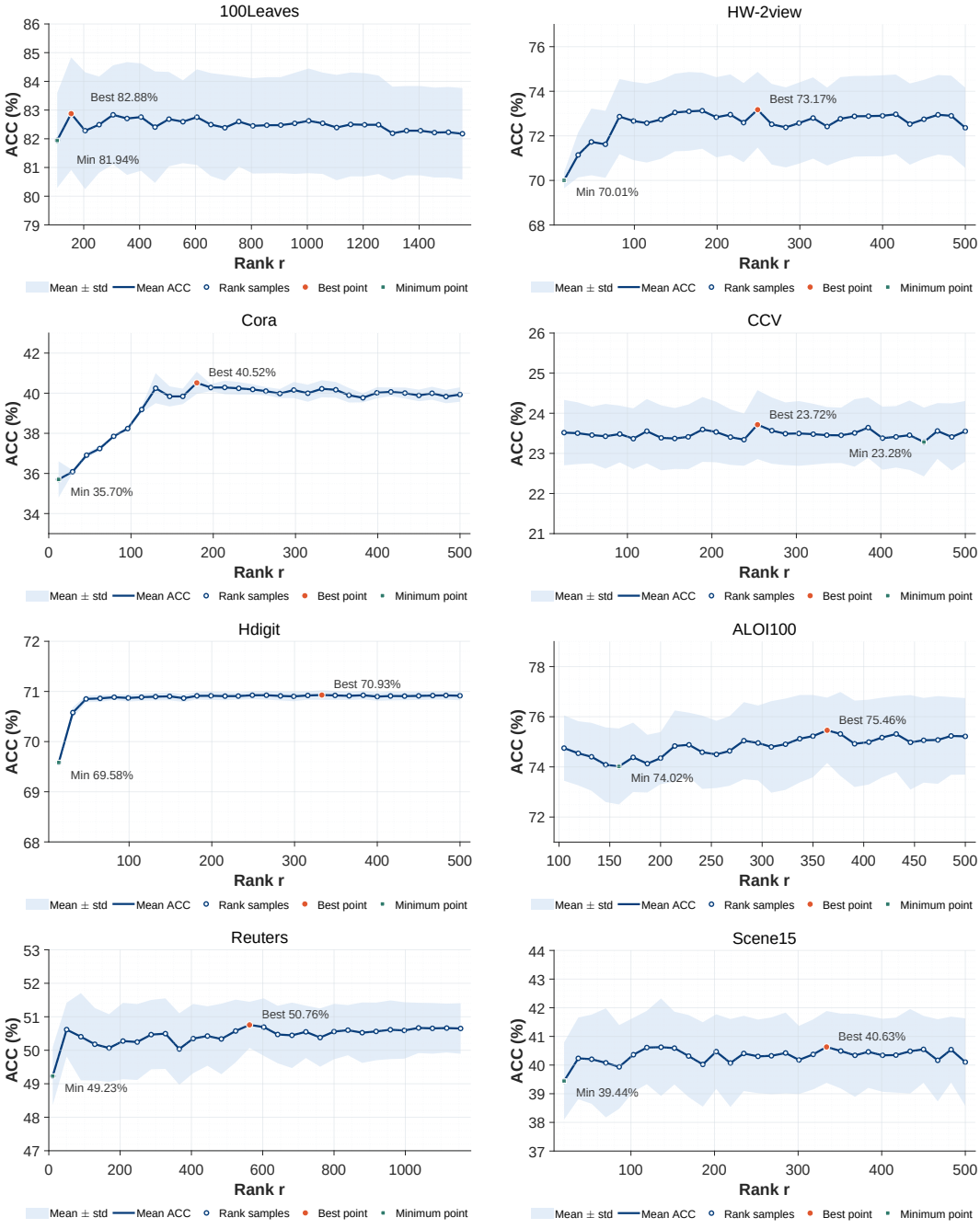



Figure 5: ACC versus compression rank $r$ on eight benchmark datasets. Solid lines and shaded regions show the mean and standard deviation, respectively. Highlighted markers in the single-method panels identify the highest and lowest observed ACC.