



Full length article

Multiple kernel clustering with local kernel reconstruction and global heat diffusion

Yan Chen ^a, Liang Du ^{b,c,*}, Peng Zhou ^d, Lei Duan ^{a,*}, Yuhua Qian ^c

^a School of Computer Science, Sichuan University, Chengdu 610065, China

^b School of Computer and Information Technology, Shanxi University, Taiyuan 030006, China

^c Institute of Big Data Science and Industry, Shanxi University, Taiyuan 030006, China

^d School of Computer Science and Technology, Anhui University, Hefei 230601, China

ARTICLE INFO

Keywords:

Multiple kernel clustering
Local kernel reconstruction
Global heat kernel diffusion
Multiple graph fusion

ABSTRACT

Multiple Kernel Clustering (MKC) is an effective approach for revealing nonlinear cluster structures in candidate kernels. However, existing MKC methods still face two key challenges. Firstly, the pairwise affinity in these methods is primarily determined by kernel similarity, disregarding the correlations among highly similar neighbors and resulting in redundant weight assignments and reduced clustering discriminability. Secondly, the direct utilization of affinity matrices overlooks high-order connections and introduces noise due to independent row-wise solving. To address these issues, we propose a novel local MKC method called LKRGDF. We begin by exploring affinity using the Local Kernel Reconstruction (LKR) model, reducing redundancy and enhancing clustering discriminability. Furthermore, we exploit the affinities with the Global heat kernel Diffusion (GD) procedure to capture long-range connections smoothly. The GD process acts as a low pass filter, focusing on small eigenvalues corresponding to top clusters. Finally, we integrate these smooth affinities within an auto-weighted Multiple Graph Fusion (MGF) framework to obtain a consensus graph. By assembling LKR, GD, and MGF in a sequential pipeline, our approach achieves the exploration and exploitation of local structures, gradually improving clustering performance while ensuring computational efficiency without the need for iterative steps. Extensive experiments on ten datasets demonstrate the superiority of our algorithm in terms of effectiveness and efficiency compared to state-of-the-art methods. The code for our method is publicly available at <https://github.com/YanChenSCU/LKRGDF-2023.git>.

1. Introduction

Kernel methods have been extensively studied to address the challenge of nonlinear data clustering. However, traditional kernel methods often require the selection of an appropriate kernel and its associated parameters, which can be difficult to determine in advance. To mitigate this issue, Multiple Kernel Clustering (MKC) has gained significant attention in recent years, allowing the combination of information from multiple base kernels for clustering purposes. While most kernel-based methods assume that all pairwise similarities between samples are reliable [1], it is acknowledged that similarity characterizations for long-range samples in high-dimensional spaces can be less reliable. Using all pairwise similarities without careful discrimination can seriously degrade the performance of kernel clustering. Moreover, prior research in clustering [2] and dimensionality reduction [3] has shown that preserving the underlying local manifold structure is more effective than preserving global pairwise similarities in unsupervised tasks.

To effectively explore and exploit the manifold structure inherent in multiple kernels, numerous local MKC approaches have been developed. These approaches leverage various techniques to capture the local structure within both the individual kernel space and the consensus kernel space. In the individual kernel space, neighbor kernels are constructed by taking the Hadamard product of the original kernel with binary neighborhood selection matrices [1,4], while affinity graphs are generated using local structure learning models [5]. In the consensus kernel space, researchers have proposed techniques to automatically learn an optimal neighborhood kernel [6] or an optimal affinity graph [7,8] to accurately characterize the manifold structure. Moreover, in the consensus kernel space, clustering can also be performed within local cliques [9–12], where a local clique is extracted for each sample [2], instead of the entire dataset.

Despite the improved performance of local MKC methods, they still face two critical challenges in exploring and exploiting local structures.

* Corresponding authors.

E-mail addresses: duliang@sxu.edu.cn (L. Du), leiduan@scu.edu.cn (L. Duan).

<https://doi.org/10.1016/j.inffus.2023.102219>

Received 14 January 2023; Received in revised form 24 October 2023; Accepted 26 December 2023

Available online 29 December 2023

1566-2535/© 2024 Elsevier B.V. All rights reserved.

Firstly, during the exploration phase, these methods tend to overlook the correlation among similar samples within a local clique. The determination of entries in the local affinity matrix heavily relies on the corresponding values within the original kernel matrix. This weighting scheme may assign higher weights to mutually redundant samples, impacting the diversity of local cliques and ultimately degrading clustering performance. Secondly, in the exploitation phase, these methods neglect the higher-order relationships across different local cliques. The sparse nature of local structure characterization within a single clique results in noisy and non-smooth coefficients in affinity graphs. The direct integration of these graphs may yield less discriminative clustering outcomes.

We present a novel sequential multiple kernel graph approach that integrates local kernel reconstruction for exploration and global heat kernel diffusion for exploitation, effectively addressing the aforementioned challenges and enhancing clustering performance. Our method consists of two sub-steps in the exploration of local structure. Firstly, we determine the neighborhood using the well-established k -nearest neighbors in the averaged kernel space. Secondly, we construct the affinity matrix by learning optimal weight assignments through local reconstruction in the kernel space, taking into account the correlation among samples within a local clique. This approach mitigates redundancy and enhances local discrimination by incorporating information from correlated samples. In the exploitation of local structure, we employ two additional sub-steps. First, we enhance candidate affinities by conducting heat kernel diffusion on each graph separately, capturing higher-order relationships among local cliques. Subsequently, we integrate these improved affinity graphs using an auto-weighted fusion model that incorporates adaptive cluster structure learning and a top- k neighbors sparse strategy. This yields a consensus graph that is more suitable for accurate clustering. It is worth noting that we adopt a separate learning procedure for the exploration and exploitation of local structure, in contrast to commonly used joint modeling strategies. This offers two advantages: firstly, the effectiveness of each step can be easily verified and validated independently; secondly, the efficiency of the algorithm is improved as it does not require iterative processes between exploration and exploitation.

In summary, the main contributions of our work can be summarized as follows.

- We introduce a novel approach to explore the optimal affinity within local cliques by using the local kernel reconstruction coefficient. This coefficient takes into account both the similarity in kernel space and the correlation among neighboring samples. This leads to improved discrimination and reduced redundancy among highly similar samples within a local clique.
- We propose a method to exploit the affinity matrix through global heat kernel diffusion. By aggregating these local cliques and capturing high-order similarities, we enhance the connectivity and reduce noise across different local cliques, resulting in a more robust and informative representation.
- We adopt a separate and consecutive learning procedure for the exploration and exploitation of local structure. This allows for independent verification of the effectiveness of each step and improves the overall efficiency of the algorithm compared to joint learning methods.

2. Related work

In this section, we present a comprehensive review of existing MKC methods and categorize them into three groups as illustrated in Fig. 1: Consensus Kernel Learning-based Early Fusion (MKC-CKL) methods, which involve the extraction of a new Positive Semi-Definite (PSD) kernel matrix; None Consensus Kernel Learning involvement early fusion (MKC-NCKL) methods; and Late Fusion (MKC-LF) methods.

The MKC-CKL methods integrate consensus kernel learning and traditional clustering paradigms within a unified framework. Firstly,

multiple kernel k -means methods integrate consensus kernel learning and Kernel K-Means (KKM) clustering within a unified framework. Consensus kernel can be obtained through different strategies, including rigorous linear combination of pre-defined kernels [13] or relaxing to optimal neighborhood kernel [14]. The linear weights can be determined at either the kernel level [15] or the sample level [16]. The matrix-induced regularization [17] can be applied to encourage diverse kernel weights. To leverage local structure, KKM can be performed on local cliques of the consensus kernel [11], allowing for handling incomplete kernels [10]. Secondly, multiple kernel subspace clustering methods integrate consensus kernel learning and self-expressiveness coefficient matrix learning of Kernel Subspace Clustering (KSC) within a unified framework. Consensus kernel can be obtained using various strategies, such as linear combinations of base kernels [18–21], optimal neighborhood kernel in proximity to the linear combined kernels [7,8,22–25], or linear combinations of neighbor kernels [1]. The consensus kernel can also be required to have a low-rank property [7,22]. The self-expressiveness coefficient matrix can be learned either in the enlarged feature space or using alternative methods [1,21]. Various techniques can be used to enhance the self-expressiveness coefficient matrix, including imposing block diagonal structures [8,19,20,23,24], acquiring sparse or low-rank structure [22], and incorporating local structure learning to improve global KSC [7,8,20]. Thirdly, pairwise kernel alignment criterion is used to guide the MKC procedure [26]. For example, Simple MKKM (SMKKM) [27] optimizes the kernel alignment criterion by maximizing the clustering matrix and minimizing the kernel coefficient through a minimization–maximization problem. The manifold structure can be captured by the performing alignment on local cliques [9,12] to capture the manifold structure. Additionally, the consensus affinity graph can be estimated by maximizing the alignment with the combined kernels [6]. In general, these MKC-CKL methods aim to accurately capture pairwise similarities by estimating a consensus kernel, resulting in improved performance through the utilization of local structures within the kernels, despite the increased computational complexity.

The MKC-NCKL approaches are characterized by the extraction of multiple graphs from candidate kernels and their integration into a consensus graph, eliminating the need for intermediate consensus PSD kernel matrix learning. For example, kernel graphs can be learned through KSC and then fused into a consensus graph using the auto-weighted graph fusion model [28]. Consensus Affinity Graph Learning (CAGL) [5] utilizes KSC with local structural learning to extract affinity graphs and learns the consensus affinity graph with sparse and Laplacian rank constraints. By incorporating pairwise and centroid-based co-regularization, the extraction of candidate graphs from kernels in a one-step learning paradigm improves the quality of the optimal affinity graph [29]. Additionally, the exploitation of the low-rank structure of a graph tensor facilitates the mutual enhancement of multiple graphs [30].

The MKC-LF approaches aim to extract multiple candidate partitions from individual kernels and integrate them to obtain a consensus cluster structure, effectively reducing the computational burden. This process involves two main sub-steps: base partition generation and fusion. These steps can be performed separately or jointly, depending on the specific approach. By generating diverse base partitions and fusing them, these approaches improve clustering performance and achieve efficient computation. Wang et al. [31] proposed an approach that maximizes the alignment between the unified kernel partition and individual partitions, resulting in an optimized consensus partition. Late Fusion with Local Kernel Alignment (LFLKA) [4] enhances base partitions by incorporating the locality enhancement kernel and maximizing alignment with the consensus partition. Projective Multiple Kernel Subspace Clustering (PMKSC) [32] constructs multiple graphs in the base partition spaces and dynamically fuses them. Late Fusion with Proxy Graph Refinement (LFPGR) [33] jointly optimizes individual kernel representations and the consensus proxy graph to achieve a

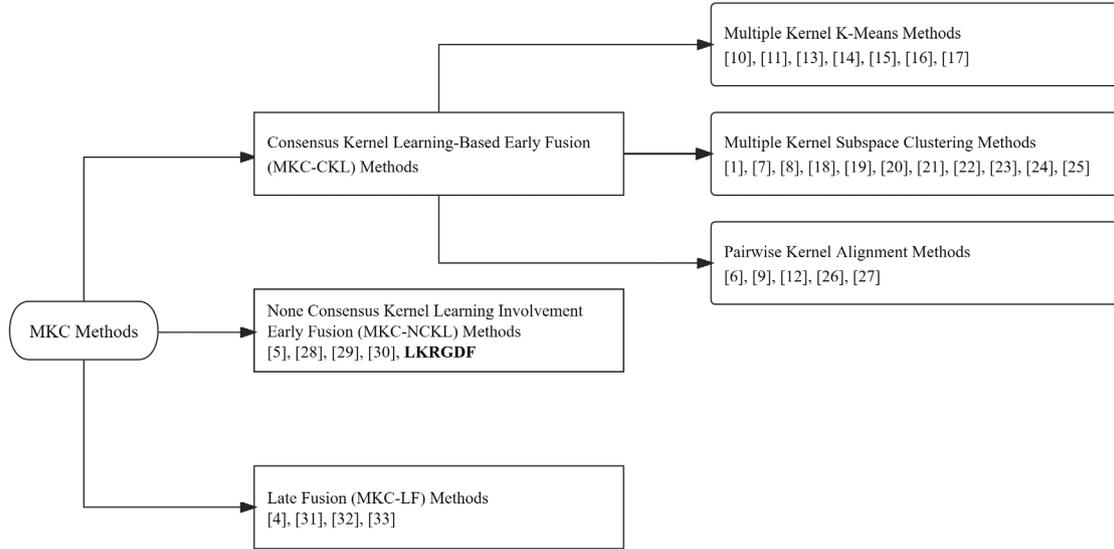


Fig. 1. The taxonomy of MKC methods.

unified formula. However, the presence of noise in partition matrices compromises the desired clear block diagonal structure in clustering, affecting overall clustering performance.

Recent advancements in MKC approaches, including both early fusion and late fusion algorithms, have underscored the significance of affinity graph learning and local clustering. Nevertheless, it is crucial to recognize that the localized pairwise similarity between two samples is predominantly shaped by their pairwise similarity in consensus or individual kernels, which often leads to redundant similarity assignments within local cliques. Additionally, these methods tend to overlook the long-range and global structural information derived from high-quality graphs.

3. The proposed formulation

Given a data set $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$, consisting of n samples in a d -dimensional input space \mathcal{X} , it is often desirable to transform these data points into a possibly infinite-dimensional reproducing kernel Hilbert space \mathcal{F} using a mapping function $\phi: \mathcal{X} \mapsto \mathcal{F}$, aiming to facilitate easier separation of the data points. The inner product between two data points in \mathcal{F} , denoted as $\langle \cdot, \cdot \rangle_{\mathcal{F}}$, is captured by the corresponding kernel function $\mathcal{K}(\cdot, \cdot)$, where $\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle_{\mathcal{F}} = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = \kappa_{ij}$. By considering a set of candidate kernel functions $\{\mathcal{K}^r(\cdot, \cdot)\}_{r=1}^m$, we can construct positive semi-definite kernel Gram matrices $\{\mathbf{K}^r\}_{r=1}^m$, with $\mathbf{K}^r \in \mathbb{R}^{n \times n}$ and $\mathbf{K}^r(i, j) = \kappa_{ij}^r$, which capture pairwise similarities across different kernel functions. The target of MKC is to effectively combine these multiple kernel matrices for clustering analysis.

3.1. Kernel graph extraction via local kernel reconstruction

There are two sub-steps for the exploration of local structure, i.e., the determination of neighborhood and the construction of affinity graph.

Since the candidate kernel matrices may not be informative for clustering, it is general to compute the average kernel matrix by combining all these kernel matrices and then robustly determining the consensus neighborhood relationship. The average kernel matrix works under a weaker assumption that only most of these base kernel matrices are informative and complementary [1]. Given a set of candidate kernels matrices $\{\mathbf{K}^r\}_{r=1}^m$, $\mathbf{K}^r \in \mathbb{R}^{n \times n}$, the average kernel \mathbf{K}_{avg} with uniform weight can be computed as, i.e.,

$$\mathbf{K}_{\text{avg}} = \frac{1}{m} \sum_{r=1}^m \mathbf{K}^r, \quad (1)$$

without using additional prior knowledge. Given the average similarity in \mathbf{K}_{avg} , the neighbors can be determined with fixed neighborhood size k or minimal similarity threshold τ [11]. Here we extract the local clique of a sample \mathbf{x}_i with its top- k nearest neighbors $\mathcal{N}_i = \{\mathbf{x}_{j_1}, \mathbf{x}_{j_2}, \dots, \mathbf{x}_{j_k}\}$ for simplicity. It can be seen that all base kernels share the same neighborhood relationship.

It has been shown that the geometric characteristics of local neighborhoods can be preserved by the local linear reconstruction weights [3, 34]. Given the neighborhood structure decided in \mathbf{K}_{avg} , we propose to perform Local Kernel Reconstruction (LKR) by minimizing the approximation error between each data point and its neighbors. Then we use its local reconstruction coefficients with neighbors in \mathbf{K}_{avg} to capture the underlying manifold structure among multiple candidate kernels. Given an arbitrary mapping function ϕ , the optimal reconstruction weight for a sample \mathbf{x}_i with its neighbors \mathcal{N}_i for the corresponding kernel Gram matrix \mathbf{K} can be determined by,

$$\begin{aligned} \min_{\bar{\mathbf{a}}_i} \quad & \|\phi(\mathbf{x}_i) - \sum_{\mathbf{x}_j \in \mathcal{N}_i} \bar{a}_j^i \phi(\mathbf{x}_j)\|^2 + \|\bar{\mathbf{a}}_i\|^2 \\ \text{s.t.} \quad & \sum_{\mathbf{x}_j \in \mathcal{N}_i} \bar{a}_j^i = 1, \bar{a}_j^i \geq 0, \end{aligned} \quad (2)$$

where $\bar{\mathbf{a}}_i = [\bar{a}_1^i, \bar{a}_2^i, \dots, \bar{a}_k^i] \in \mathbb{R}^{k \times 1}$ is the local reconstruction coefficient of \mathbf{x}_i . We introduce a vector $\mathbf{k}_i = [\kappa_{ij_1}, \kappa_{ij_2}, \dots, \kappa_{ij_k}] \in \mathbb{R}^{k \times 1}$ to denote the kernel function value between \mathbf{x}_i and its neighbors, and define a matrix $\mathbf{E}^i \in \{0, 1\}^{n \times k}$ to indicate the selection of neighbors for \mathbf{x}_i . The (j, j') -entry of \mathbf{E}^i is determined as follows:

$$\mathbf{E}^i(j, j') = \begin{cases} 1, & \text{if } \mathbf{x}_j \in \mathcal{N}_i, j' \text{ is the index of } \mathbf{x}_j \text{ in } \mathcal{N}_i, \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

Then the induced local kernel for the i th sample can be denoted as $\mathbf{K}^i = (\mathbf{E}^i)^T \mathbf{K} \mathbf{E}^i \in \mathbb{R}^{k \times k}$, which is a sub-matrix of a kernel matrix \mathbf{K} corresponding to the neighbors of \mathbf{x}_i . With these newly introduced variables, the problem in Eq. (2) can be reformulated as follows,

$$\begin{aligned} \min_{\bar{\mathbf{a}}_i} \quad & \bar{\mathbf{a}}_i^T \mathbf{K}^i \bar{\mathbf{a}}_i - 2\bar{\mathbf{a}}_i^T \mathbf{k}_i + \bar{\mathbf{a}}_i^T \bar{\mathbf{a}}_i \\ & = \sum_{u=1}^k \sum_{v=1}^k \bar{a}_u^i \kappa_{uv}^i \bar{a}_v^i - 2 \sum_{u=1}^k \bar{a}_u^i \kappa_{u}^i + \sum_{u=1}^k (\bar{a}_u^i)^2 \\ \text{s.t.} \quad & \bar{\mathbf{a}}_i^T \mathbf{1}_k = 1, \bar{\mathbf{a}}_i \geq \mathbf{0}_k, \end{aligned} \quad (4)$$

where u, v are the index of two neighbors of \mathbf{x}_j , \mathbf{x}_j in \mathcal{N}_i , \bar{a}_u^i and κ_{uv}^i represent the u th element in the vectors $\bar{\mathbf{a}}_i$ and \mathbf{k}_i , κ_{uv}^i is the (u, v) -entry of \mathbf{K}^i . These terms offer distinct advantages within the context

of our framework. The first term serves to enhance the diversity of neighbors within a local clique and reduce redundancy. Specifically, it prevents the simultaneous assignment of higher weights to two highly similar and correlated neighbors of a given data point, which can lead to a more balanced weight distribution. The second term is instrumental in preserving the local manifold structure in the kernel space by maximizing the alignment between the coefficient and the local kernel similarity. It ensures that higher similarity is associated with higher weight assignments, capturing the underlying structure effectively. Finally, the third term acts as a prior that contributes to balancing the uniformity of neighborhood assignments, ensuring a more equitable distribution of weights across neighbors.

Given the LKR coefficients $\{\tilde{a}_{ij}\}_{j=1}^n$ for all samples, we introduce a non-negative affinity matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, which can be fulfilled as follows,

$$a_{ij} = \begin{cases} \tilde{a}_{ij}^j, & j' \text{ is the index of } \mathbf{x}_j \text{ in } \mathcal{N}_i, \\ 0, & \text{otherwise,} \end{cases} \quad (5)$$

to capture the local manifold structure in a specific kernel space \mathbf{K} . It can be verified that $a_{ij} \geq 0$ and $\sum_{j=1}^n a_{ij} = 1$. Totally, we extract a set of affinity matrices $\{\mathbf{A}^r\}_{r=1}^m$ to capture the candidate manifold structure from kernels $\{\mathbf{K}^r\}_{r=1}^m$. The important manifold structure within local neighborhoods can be characterized by the local reconstruction weights \mathbf{A}^r , which in turn can be used to guide the multiple kernel fusion and the final clustering result. The experimental results in Section 5.5 also support our analysis.

3.2. Kernel graph smoothing via global heat kernel diffusion

It can be seen that the coefficients in Eq. (5) are obtained row-by-row independently. As a result, the global interconnection between these rows are neglected. Before integrating these local affinities, we introduce high-order connections through the heat kernel diffusion procedure to fulfill this gap. The immediate information could be characterized by passing messages between neighboring samples. For simplicity and clarity of representation, we will use the symbol \mathbf{A} to refer to an arbitrary graph within the set $\{\mathbf{A}^r\}_{r=1}^m$, which is obtained from the aforementioned LKR model. Given an asymmetric affinity graph \mathbf{A} , we first introduce the corresponding symmetric matrix $\tilde{\mathbf{A}} = (\mathbf{A} + \mathbf{A}^T)/2$, the diagonal degree matrix \mathbf{D} where its diagonal element $\mathbf{D}(i, i) = \sum_{j=1}^n \tilde{a}_{ij}$, the symmetric transition matrix $\mathbf{Z} = \mathbf{D}^{-\frac{1}{2}} \tilde{\mathbf{A}} \mathbf{D}^{-\frac{1}{2}}$, and the normalized graph Laplacian matrix $\mathbf{L} = \mathbf{I} - \mathbf{Z}$. The second order relationship can be calculated by the matrix multiplication, i.e., $\mathbf{Z}^2 = \mathbf{Z}\mathbf{Z}$. Similarly, a larger range of immediate higher-order information can be recursively obtained with a series of matrices $\mathbf{Z}^3, \mathbf{Z}^4, \dots, \mathbf{Z}^\infty$. Finally, we get the diffusion matrix by aggregating all these matrices in the form of,

$$\mathbf{G} = \sum_{t=0}^{\infty} w_t \mathbf{Z}^t, \quad (6)$$

where w_t is the immediate weighting coefficient, and \mathbf{Z}^0 is the uninformative identity matrix.

In this paper, we adopt the heat kernel diffusion procedure [35] to determine the weight of aggregation, i.e.,

$$w_t = e^{-\eta} \frac{\eta^t}{t!}, \quad (7)$$

where η is a non-negative value of temperature to control the decay speed. It can be verified that $\sum_{t=0}^{\infty} w_t = 1, w_t \in [0, 1]$. The heat kernel diffusion satisfies the heat equation and can be viewed as describing the flow of heat across the edge of a graph with time, where the rate of flow is determined by the Laplacian of the graph [36]. By merging Eqs. (6) and (7), we get the Global heat kernel Diffusion (GD) matrix as,

$$\mathbf{G} = e^{-\eta} \sum_{t=0}^{\infty} \frac{\eta^t}{t!} \mathbf{Z}^t = \sum_{t=0}^{\infty} \frac{(-\eta)^t}{t!} \mathbf{L}^t = \exp^{-\eta \mathbf{L}}, \quad (8)$$

where the last two equations hold according to the definition of the matrix exponential. Based on the above aggregation, we have the following observations. On the one hand, the heat kernel is dominated by the local connectivity structure encoded in graph Laplacian \mathbf{L} when t is small. On the other hand, the heat kernel is governed by the global structure of the graph when t increases [37].

From the perspective of spectral graph theory, the smoothness of a graph can be captured by the following quadratic function [38] over nodes,

$$\begin{aligned} & \sum_{i,j=1}^n \tilde{a}_{ij} \left(\frac{\mathbf{u}_i}{\sqrt{\mathbf{D}(i, i)}} - \frac{\mathbf{u}_j}{\sqrt{\mathbf{D}(j, j)}} \right)^2 \\ &= \text{tr} \left(\mathbf{U}^T (\mathbf{I} - \mathbf{D}^{-\frac{1}{2}} \tilde{\mathbf{A}} \mathbf{D}^{-\frac{1}{2}}) \mathbf{U} \right) = \text{tr} \left(\mathbf{U}^T (\mathbf{I} - \mathbf{Z}) \mathbf{U} \right) \\ &= \sum_{i=1}^n \mathbf{u}_i^T \mathbf{L} \mathbf{u}_i = \sum_{i=1}^n \lambda_i, \end{aligned} \quad (9)$$

where $\mathbf{L} = \mathbf{U}\mathbf{A}\mathbf{U}^T$ is the eigen decomposition of the Laplacian \mathbf{L} , \mathbf{U} is the eigenvector matrix, $\mathbf{A} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$ is the diagonal eigenvalue matrix with eigenvalues sorted in ascending order $0 = \lambda_1 \leq \lambda_2 \leq \lambda_n$. Therefore the smoothness of the graph is determined by the summation of eigenvalues of the corresponding graph Laplacian and dominated by the higher ones. Given an eigenvector and eigenvalue pair $(\mathbf{u}_i, \lambda_i)$ of \mathbf{L} , it can be verified that \mathbf{u}_i is also an eigenvector of \mathbf{G} with new eigenvalue $\exp^{-\eta \lambda_i}$. The eigen decomposition of \mathbf{G} can be further written as,

$$\mathbf{G} = \mathbf{U} \left(\sum_{t=0}^{\infty} \frac{(-\eta)^t}{t!} \mathbf{A}^t \right) \mathbf{U}^T = \mathbf{U} \tilde{\mathbf{A}} \mathbf{U}^T, \quad (10)$$

where $\tilde{\mathbf{A}} = \text{diag}(\exp^{-\eta \lambda_1}, \exp^{-\eta \lambda_2}, \dots, \exp^{-\eta \lambda_n})$. It indicates that the heat kernel diffusion matrix \mathbf{G} has the same eigenvectors of \mathbf{L} with exponential decay of its eigenvalues.

From the perspective of the graph signal process, a graph filter $\mathcal{H}(\mathbf{L})$ is defined as $\mathcal{H}(\mathbf{L}) = \sum_{t=0}^{\bar{t}} h_t \mathbf{L}^t$, where \bar{t} is the order of graph filter. Given a graph signal $\hat{\mathbf{x}} \in \mathbb{R}^{n \times 1}$, the response on graph \mathbf{G} can be written as a linear combination of the input signal, i.e., $\hat{\mathbf{y}} = \mathcal{H}(\mathbf{L})\hat{\mathbf{x}} = \mathbf{G}\hat{\mathbf{x}}$. Their frequency domain representation can be computed by the Graph Fourier Transform (GFT) in the form of $\tilde{\mathbf{x}} = \mathbf{U}^T \hat{\mathbf{x}} = [\mathbf{u}_1^T \hat{\mathbf{x}}, \mathbf{u}_2^T \hat{\mathbf{x}}, \dots, \mathbf{u}_n^T \hat{\mathbf{x}}]$ and $\tilde{\mathbf{y}} = \mathbf{U}^T \hat{\mathbf{y}} = [\mathbf{u}_1^T \hat{\mathbf{y}}, \mathbf{u}_2^T \hat{\mathbf{y}}, \dots, \mathbf{u}_n^T \hat{\mathbf{y}}]$. A graph filter in the frequency domain can be defined as $\tilde{\mathbf{h}} = [\tilde{h}_1, \tilde{h}_2, \dots, \tilde{h}_n]$, where $\tilde{h}_i = \sum_{t=0}^{\bar{t}} h_t \lambda_i^t$ is the generating function of the graph filter. The component of $\tilde{\mathbf{y}}$ in frequency domain can be written as $\tilde{y}_i = \tilde{h}(\lambda_i) \tilde{x}_i = \sum_{t=0}^{\bar{t}} h_t \lambda_i^t \tilde{x}_i = \sum_{t=0}^{\infty} \frac{(-\eta)^t}{t!} \lambda_i^t \tilde{x}_i = \exp^{-\eta \lambda_i} \tilde{x}_i$. To further demonstrate the property of \mathbf{G} , we present the definition of the low-pass graph filter as follows [39],

Definition 1. Given a generating function $\tilde{h}_i = \sum_{t=0}^{\bar{t}} h_t \lambda_i^t$, a graph filter $\mathcal{H}(\mathbf{L})$ is a (i, η) low-pass graph filter [40] if the low pass ratio η satisfies

$$\eta = \frac{\max\{\tilde{h}(\lambda_{i+1}), \tilde{h}(\lambda_{i+2}), \dots, \tilde{h}(\lambda_n)\}}{\min\{\tilde{h}(\lambda_1), \tilde{h}(\lambda_2), \dots, \tilde{h}(\lambda_i)\}} \in [0, 1). \quad (11)$$

Since the eigenvalues λ_i and the generating function \tilde{h}_i are monotonic, the i th pass ratio of \mathbf{G} can be computed by,

$$\eta_i = \frac{e^{-\eta \lambda_{i+1}}}{e^{-\eta \lambda_i}} = e^{-\eta(\lambda_{i+1} - \lambda_i)} \in [0, 1). \quad (12)$$

It indicates that the heat kernel diffusion matrix \mathbf{G} can be seen as a low-pass graph filter.

Instead of integrating these local affinity matrices $\{\mathbf{A}^r\}_{r=1}^m$ directly, we first exploit them with GD as demonstrated in Eq. (10) and generate a set of enhanced graphs $\{\mathbf{G}^r\}_{r=1}^m$. The heat kernel diffusion smooths out the neighborhood by bridging the local and global structure via the aggregation of intermediate higher-order information. The heat kernel diffusion filters out the larger eigenvalues corresponding to noisy fine details via exponential decay transformation while amplifying the smaller eigenvalues corresponding to the top clusters in the graph. Consequently, the heat kernel diffusion-induced graph smoothing is helpful in reducing undesirable distortions and noise while preserving crucial manifold structure. Thus, the clustering performance can be further improved by the newly introduced diffusion procedure.

3.3. Multiple graph fusion

By leveraging the intrinsic and discriminant structure captured within the multiple candidate graphs, the original problem of MKC is transformed into the task of clustering with multiple graphs. With the set of well-diffused and connected candidate graphs $\{\mathbf{G}^r\}_{r=1}^m$ obtained through the previous steps, our objective is to derive the final consensus clustering result within the framework of multiple graph clustering. In this paper, we propose a straightforward yet effective method for learning the consensus graph by seeking the optimal graph in the vicinity of each individual graph, while minimizing the approximation errors,

$$\begin{aligned} \min_{\mathbf{S}, \mathbf{p}} \quad & \sum_{r=1}^m \frac{1}{p_r} \|\mathbf{S} - \mathbf{G}^r\|^2 \\ \text{s.t.} \quad & \mathbf{S}\mathbf{1}_n = \mathbf{1}_n, \mathbf{S} \geq 0, \sum_{r=1}^m p_r = 1, \mathbf{p} \geq \mathbf{0}_m, \end{aligned} \quad (13)$$

where $\mathbf{S} \in \mathbb{R}^{n \times n}$ is the consensus graph and $\mathbf{p} \in \mathbb{R}^{m \times 1}$ is the weights of graphs.

In order to improve the suitability of the estimated consensus similarity matrix \mathbf{S} for clustering purposes, we incorporate a commonly used block diagonal constraint on \mathbf{S} . Specifically, we enforce the constraint $\text{Rank}(\mathbf{L}_{\mathbf{S}}) = n - c$, where $\mathbf{L}_{\mathbf{S}}$ denotes the Laplacian matrix derived from \mathbf{S} . This constraint encourages an ideal cluster assignment within \mathbf{S} [41]. Furthermore, we impose sparsity on the consensus graph to enhance its locality and reduce noise in the connections. Specifically, we retain only the top k neighbors for each row of the consensus graph. This sparsity constraint helps to focus on the most relevant and informative connections within the graph. In summary, the final formulation of our Multiple Graph Fusion (MGF) model can be expressed as follows, taking into account both the block diagonal constraint and the sparsity constraint:

$$\begin{aligned} \min_{\mathbf{S}, \mathbf{p}} \quad & \sum_{r=1}^m \frac{1}{p_r} \|\mathbf{S} - \mathbf{G}^r\|^2 \\ \text{s.t.} \quad & \mathbf{S}\mathbf{1} = \mathbf{1}, \mathbf{S} \geq 0, \text{Rank}(\mathbf{L}_{\mathbf{S}}) = n - c, \\ & \sum_{r=1}^m p_r = 1, \mathbf{p} \geq \mathbf{0}_m, |\hat{s}_i|_0 = k, \forall i, \end{aligned} \quad (14)$$

where \hat{s}_i represents the i th row of matrix \mathbf{S} . In the above formulation, the objective is to find the consensus graph \mathbf{S} that minimizes the approximation errors with respect to the individual graphs, while satisfying the block diagonal and sparsity constraints. Once the consensus graph \mathbf{S} is obtained, we can derive the final clustering results using graph cut algorithms, such as spectral clustering.

We now summarize the details of the proposed method. In the stage of kernel structure exploration, we propose to construct local affinity matrices using the coefficients of the local kernel reconstruction model. It conducts reconstruction on the local clique of each sample within the kernel space, thereby reducing the redundancy of similar neighbors and enhancing the discriminatory power for clustering. During the kernel structure exploitation stage, we propose to establish long-range connections through global heat kernel diffusion. This process leverages the low-pass filter property to smooth out large eigenvalues, thereby focusing on smaller eigenvalues that correspond to the top clusters. We then integrate these smoothed affinities within an auto-weighted multiple graph fusion framework. Additionally, the exploration and exploitation of kernel structure are assembled into a pipeline that unfolds through three consecutive steps. In order to better understand the proposed method, the block diagram is illustrated in Fig. 2.

4. Optimization and analysis

In this section, we present an efficient algorithm associated with the above three steps problem.

4.1. Obtaining $\{\mathbf{A}^r\}_{r=1}^m$

For the r th single kernel \mathbf{K}^r , each row of the LKR coefficient matrix \mathbf{A}^r can be derived by solving the quadratic problem with a simplex constraint, as described in Eq. (4), involving only k variables. We can obtain the optimal solution for Eq. (4) using off-the-shelf quadratic problem solvers. Subsequently, we can assemble the entire LKR coefficient matrix \mathbf{A}^r according to Eq. (5).

4.2. Obtaining $\{\mathbf{G}^r\}_{r=1}^m$

Given the LKR coefficient matrices $\{\mathbf{A}^r\}_{r=1}^m$, the immediate high-order information can be captured through the diffusion process as shown in Eq. (6). The close form solution can be computed from Eq. (8). Furthermore, a rapid approximation of the diffused graph can be achieved by truncating the top \bar{t} diffusion steps recursively, wherein the sparse matrix multiplication can be performed solely on the non-zero entries.

4.3. Obtaining \mathbf{S}

Given multiple diffused graphs $\{\mathbf{G}^r\}_{r=1}^m$, we first relax the exact rank constraint in Eq. (14) according to the Ky Fan's Theorem [41] and get the equivalent problem as follows

$$\begin{aligned} \min_{\mathbf{S}, \mathbf{p}, \mathbf{F}} \quad & \sum_{r=1}^m \frac{1}{p_r} \|\mathbf{S} - \mathbf{G}^r\|^2 + \lambda \text{tr}(\mathbf{F}^T \mathbf{L}_{\mathbf{S}} \mathbf{F}) \\ \text{s.t.} \quad & \mathbf{S}\mathbf{1} = \mathbf{1}, \mathbf{S} \geq 0, |\hat{s}_i|_0 = k, \forall i, \sum_{r=1}^m p_r = 1, \mathbf{p} \geq \mathbf{0}_m, \mathbf{F}^T \mathbf{F} = \mathbf{I}, \end{aligned} \quad (15)$$

where λ is a regularization parameter. The above problem has three variables, i.e., \mathbf{F} , \mathbf{p} and \mathbf{S} . The optimal solution can be obtained by the following iterative coordinate descent procedure with three sub-steps.

Optimizing \mathbf{F} given \mathbf{p} and \mathbf{S} . The optimization problem w.r.t. \mathbf{F} can be written as,

$$\min_{\mathbf{F}} \quad \text{tr}(\mathbf{F}^T \mathbf{L}_{\mathbf{S}} \mathbf{F}), \quad \text{s.t.} \quad \mathbf{F}^T \mathbf{F} = \mathbf{I}. \quad (16)$$

The optimal solution of \mathbf{F} is the top c eigenvectors of $\mathbf{L}_{\mathbf{S}}$ corresponding to the top c smallest eigenvalues.

Optimizing \mathbf{p} given \mathbf{F} and \mathbf{S} . By introducing $\mathbf{e} \in \mathbb{R}^{m \times 1}$ with $e_r = \|\mathbf{S} - \mathbf{G}^r\|^2$, the problem w.r.t. \mathbf{p} becomes,

$$\min_{\mathbf{p}} \quad \sum_{r=1}^m \frac{e_r}{p_r}, \quad \text{s.t.} \quad \sum_{r=1}^m p_r = 1, \mathbf{p} \geq \mathbf{0}_m. \quad (17)$$

According to the Cauchy-Schwarz inequality, we have

$$\sum_{r=1}^m \frac{e_r}{p_r} = \left(\sum_{r=1}^m \frac{e_r}{p_r} \right) \left(\sum_{r=1}^m p_r \right) \geq \left(\sum_{r=1}^m \sqrt{e_r} \right)^2. \quad (18)$$

The equality in Eq. (18) holds when $p_r \propto \sqrt{e_r}$. The close form solution [42] of \mathbf{p} can be written as,

$$p_r = \frac{\sqrt{e_r}}{\sum_{j=1}^m \sqrt{e_j}}. \quad (19)$$

Optimizing \mathbf{S} given \mathbf{F} and \mathbf{p} . The rest problem w.r.t. \mathbf{S} can be written as,

$$\begin{aligned} \min_{\mathbf{S}} \quad & \sum_{r=1}^m \frac{1}{p_r} \|\mathbf{S} - \mathbf{G}^r\|^2 + \frac{\lambda}{2} \sum_{i=1}^n \sum_{j=1}^n \|\mathbf{f}_i - \mathbf{f}_j\|^2 s_{ij} \\ \text{s.t.} \quad & \mathbf{S}\mathbf{1} = \mathbf{1}, \mathbf{S} \geq 0, |\hat{s}_i|_0 = k, \forall i. \end{aligned} \quad (20)$$

The above problem can be decomposed into n independent sub-problems w.r.t. each row of \mathbf{S} (i.e., \hat{s}_i) in the form of,

$$\min_{\hat{s}_i} \quad \sum_{r=1}^m \frac{1}{p_r} \|\hat{s}_i - \hat{\mathbf{g}}_i^r\|^2 + \frac{\lambda}{2} \hat{s}_i \hat{\mathbf{q}}_i^T \quad (21)$$

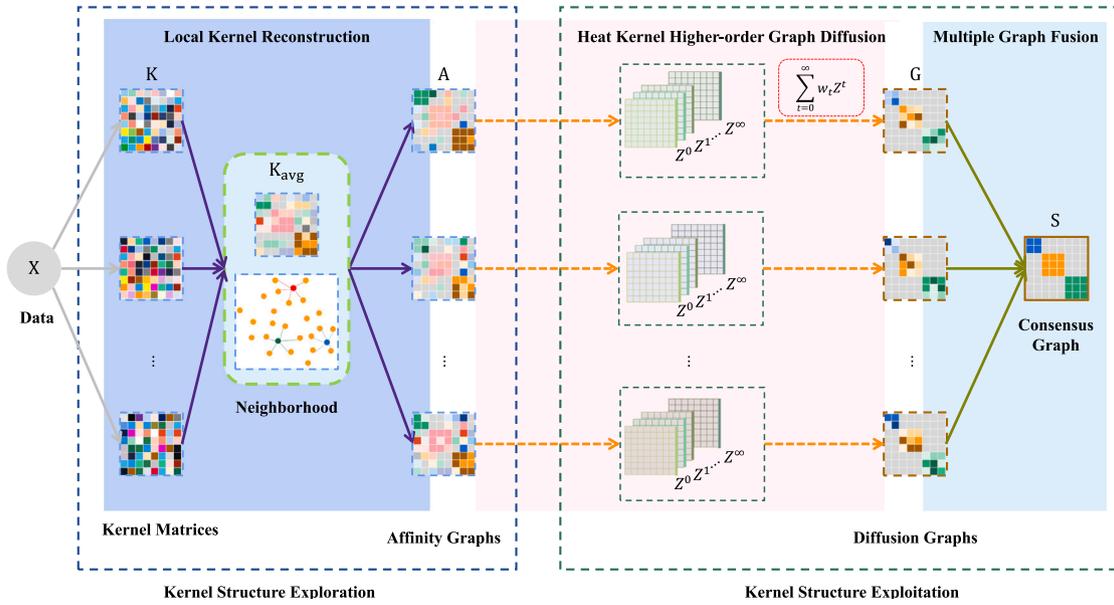


Fig. 2. The framework of the proposed LKRGDF method.

$$\text{s.t. } \hat{\mathbf{s}}_i^T \mathbf{1}_n = 1, \hat{\mathbf{s}}_i \geq 0, |\hat{\mathbf{s}}_i|_0 = k,$$

where $\hat{\mathbf{q}}_i \in \mathbb{R}^{1 \times n}$ and the j th element is given by $\hat{q}_j^i = \|\mathbf{f}_i - \mathbf{f}_j\|^2$, $\hat{\mathbf{g}}_i^r$ is the i -row of \mathbf{G}^r . Eq. (21) can be simplified as,

$$\min_{\hat{\mathbf{s}}_i} \|\hat{\mathbf{s}}_i - \hat{\mathbf{b}}_i\|^2 \quad (22)$$

$$\text{s.t. } \hat{\mathbf{s}}_i^T \mathbf{1}_n = 1, \hat{\mathbf{s}}_i \geq 0, |\hat{\mathbf{s}}_i|_0 = k,$$

where $\hat{\mathbf{b}}_i = (\sum_{r=1}^m \frac{1}{p_r} \hat{\mathbf{g}}_i^r - \frac{\lambda}{4} \hat{\mathbf{q}}_i) / (\sum_{j=1}^m \frac{1}{p_j})$. The above problem is the euclidean projection on a simplex with cardinality constraint. Without loss of generality, suppose $\hat{\mathbf{b}}_i = [b_{i1}, b_{i2}, \dots, b_{in}]$ are sorted in descending order. If there are only k non-zero elements in $\hat{\mathbf{s}}_i$ as required, we have $[s_{i1}, s_{i2}, \dots, s_{ik}] \geq 0$ and $s_{ij} = 0$ if $j > k$ according to the induced KKT condition of Eq. (22). Then the above problem w.r.t n variables can be further reduced to a small scale problem with only k variables, which can be solved by off-the-shelf algorithm efficiently [43]. To control the rank of \mathbf{L}_S in Eq. (15), we initialize $\lambda = 1$ and adjust it automatically during iterations, i.e., double it if $\text{Rank}(\mathbf{L}_S) > n - c$ or half it if $\text{Rank}(\mathbf{L}_S) < n - c$.

The proposed three-step optimization procedure is outlined in Algorithm 1. k is the neighborhood size and η is the diffusion parameter. Besides, the cluster number c should also be specified in advance.

Algorithm 1 Algorithm of the proposed method

Input: Multiple kernels $\{\mathbf{K}^r\}_{r=1}^m$, the neighborhood size k , the diffusion parameter η , the cluster number c .

Compute multiple local kernel reconstruction coefficient matrices $\{\mathbf{A}^r\}_{r=1}^m$ according to Eq. (4) and Eq. (5);

Compute multiple graph diffusion matrices $\{\mathbf{G}^r\}_{r=1}^m$ according to Eq. (8);

Initialize $\mathbf{S} = \frac{1}{m} \sum_{r=1}^m \mathbf{G}^r$, $\mathbf{p} = \frac{1}{m} \mathbf{1}_m$, $\lambda = 1$;

repeat

 Update \mathbf{F} according to Eq. (16);

 Update \mathbf{p} according to Eq. (19);

 Update each row $\hat{\mathbf{s}}_i$ according to Eq. (22);

 Update λ automatically;

until Converges

Output: Perform spectral clustering on consensus affinity matrix $\frac{\mathbf{S} + \mathbf{S}^T}{2}$.

Table 1

Computational complexity of the proposed method.

Neighborhood determination	LKR	GD	MGF	Total
$\mathcal{O}(mn^2 + n^2 \log k)$	$\mathcal{O}(mnk^3)$	$\mathcal{O}(m\bar{n}^2 k)$	$\mathcal{O}(n^2 c + mn^2 + nk^3)\bar{t}$	$\mathcal{O}(n^2)$

4.4. Complexity analysis

Now, we analyze the time complexity of the proposed method. The cost of computing the averaged kernel matrix \mathbf{K}_{avg} in Eq. (1) is $\mathcal{O}(mn^2)$. The finding of top k -nearest neighbors for each sample costs $\mathcal{O}(n \log k)$, and the total cost for all samples is $\mathcal{O}(n^2 \log k)$. The cost of computing the reconstruction weight for each sample is $\mathcal{O}(k^3)$, and the total cost for all samples in a single kernel is $\mathcal{O}(nk^3)$. Thus, the total computational complexity of computing these reconstruction matrices $\{\mathbf{A}^r\}_{r=1}^m$ is $\mathcal{O}(mn^2 + n^2 \log k + mnk^3)$. The general cost of matrix exponential for graph diffusion is $\mathcal{O}(n^3)$. However, the polynomial heat kernel diffusion can be efficiently approximated by the aggregated matrix products as in Eq. (6) with sparse \mathbf{Z} , having only k nonzero elements in each row. Therefore the time complexity of computing these graph diffusion can be further reduced to $\mathcal{O}(m\bar{n}^2 k)$, where \bar{t} is the order of diffusion and $\bar{t} \ll n$. The cost of solving problem in Eq. (16) is $\mathcal{O}(n^2 c)$ when partial SVD is used. The computing of Eqs. (19) and (22) is $\mathcal{O}(mn^2 + nk^3)$. The total cost for solving Eq. (14) is $\mathcal{O}((n^2 c + mn^2 + nk^3)\bar{t})$, where \bar{t} is the number of iterations. Therefore, the total cost of the three-step optimization procedure can be summarized as $\mathcal{O}(n^2 + mnk^3 + m\bar{n}^2 k + (n^2 c + mn^2 + nk^3)\bar{t})$. Since $m, k, c \ll n$, \bar{t} , and \bar{t} are usually small (often within 10 iterations), the total complexity can be denoted as $\mathcal{O}(n^2)$ in general. Compared with many early fusion MKC methods [6,11,19,32] with $\mathcal{O}(n^3)$, our method is computationally efficient, as demonstrated in Table 1 and Table 4.

4.5. Connection to previous methods

In this subsection, we carefully investigate several recent MKC approaches from the perspective of single kernel graph structure exploration.

- KSC. A variety of MKC methods [1,7,23,32,33,44] extend the Subspace Clustering (SC) model [45] into the kernel space. This adaptation, referred to Kernel Subspace Clustering (KSC), aims to

extract the global reconstruction affinity by solving the following problem,

$$\begin{aligned} \min_{\mathbf{S}} \quad & \|\phi(\mathbf{X}) - \phi(\mathbf{X})\mathbf{S}\|^2 + \beta\|\mathbf{S}\|^2 \\ & = \text{tr}(\mathbf{S}^T \mathbf{K} \mathbf{S}) - 2\text{tr}(\mathbf{K} \mathbf{S}) + \beta \text{tr}(\mathbf{S}^T \mathbf{S}) \\ \text{s.t.} \quad & \text{diag}(\mathbf{S}) = \mathbf{0}, \mathbf{S} \geq 0. \end{aligned} \quad (23)$$

- **NK- k .** Zhou et al. [1] proposes to identify the top- k neighbors in the average kernel matrix \mathbf{K}_{avg} . We denote the neighbor indicator matrix for a sample i as $\tilde{\mathbf{E}}^i \in \{0, 1\}^{n \times n}$. Here, $\tilde{\mathbf{E}}^i(j, j') = 1$ if \mathbf{x}_j and $\mathbf{x}_{j'}$ are both neighbors of sample \mathbf{x}_i , and $\tilde{\mathbf{E}}^i(j, j') = 0$ otherwise. The Neighbor Kernel determined by k (NK- k) can be obtained by $\tilde{\mathbf{K}} = \sum_{i=1}^n \tilde{\mathbf{E}}^i \circ \mathbf{K}$, where \circ is the Hadamard product.
- **NK- τ .** Liu et al. [11] proposes to determine the neighborhood using a threshold τ in \mathbf{K}_{avg} . We denote the neighbors for a sample i as $\Omega_i = \{j | \kappa_{ij} \geq \tau\}$ and the corresponding selection matrix as $\tilde{\mathbf{E}}^i \in \{0, 1\}^{n \times \mu_i}$, where μ_i is the size of Ω_i . Here, $\tilde{\mathbf{E}}^i(j, j') = 1$ if \mathbf{x}_j and $\mathbf{x}_{j'}$ are both neighbors of sample \mathbf{x}_i , and $\tilde{\mathbf{E}}^i(j, j') = 0$ otherwise. The induced local kernel for the i th sample can be formulated as $\tilde{\mathbf{K}}^i = \tilde{\mathbf{E}}^{iT} \mathbf{K} \tilde{\mathbf{E}}^i \in \mathbb{R}^{\mu_i \times \mu_i}$. The KKM on Neighbor Kernel determined by τ (NK- τ) can be obtained by solving the following optimization problem [9].

$$\min_{\mathbf{H}} \quad \sum_{i=1}^n \tilde{\mathbf{E}}^{iT} \mathbf{K} \tilde{\mathbf{E}}^i (\mathbf{I}_{\mu_i} - \mathbf{H}^i \mathbf{H}^{iT}), \quad \text{s.t.} \quad \mathbf{H}^T \mathbf{H} = \mathbf{I}, \mathbf{H}^i = \tilde{\mathbf{E}}^{iT} \mathbf{H}. \quad (24)$$

- **KLSL.** LSWMKC [6] utilizes the Kernel Local Structure Learning (KLSL) model, which does not involve adaptive structure learning. The KLSL model can be formulated as follows:

$$\min_{\mathbf{S}} \quad -\text{tr}(\mathbf{K} \mathbf{S}) + \beta \text{tr}(\mathbf{S}^T \mathbf{S}), \quad \text{s.t.} \quad \mathbf{S} \mathbf{1} = \mathbf{1}, \mathbf{S} \geq 0. \quad (25)$$

- **KALSLS.** The Adaptive Local Structure Learning (ALSLS) method has been widely utilized in clustering [41]. SPMKC [8] extends ALSLS into the kernel space and obtains KALSLS in the form of,

$$\min_{\mathbf{S}} \quad -\text{tr}(\mathbf{K} \mathbf{S}) + \beta \text{tr}(\mathbf{S}^T \mathbf{S}), \quad (26)$$

$$\text{s.t.} \quad \mathbf{S} \mathbf{1} = \mathbf{1}, \mathbf{S} \geq 0, \text{rank}(\mathbf{L}_{\mathbf{S}}) = n - c. \quad (27)$$

- **LKSC.** CAGL [5] integrates the above KSC and KALSLS models to capture both the global and local structure, resulting in the localized KSC model denoted as LKSC. The formulation of LKSC is as follows:

$$\min_{\mathbf{S}} \quad \text{tr}(\mathbf{S}^T \mathbf{K} \mathbf{S}) - \alpha \text{tr}(\mathbf{K} \mathbf{S}) + \beta \|\mathbf{S}\|^2, \quad \text{s.t.} \quad \mathbf{S} \mathbf{1} = \mathbf{1}, \mathbf{S} \geq 0. \quad (28)$$

CAGL also emphasizes the importance of manually adjusting the weight of the second term of Eq. (28) for improved clustering analysis.

- **LNWKR.** The Nadaraya–Watson kernel estimator is a popular non-parametric regression technique for nonlinear regression [46]. Given neighbors \mathcal{N}_i of sample \mathbf{x}_i , the target label y_i can be computed by $y_i = (\sum_{\mathbf{x}_{j'} \in \mathcal{N}_i} \kappa_{ij'} y_{j'}) / (\sum_{\mathbf{x}_{j'} \in \mathcal{N}_i} \kappa_{ij'})$, and a local affinity matrix \mathbf{S} can be filled by the Localized Nadaraya–Watson Kernel Regression (LNWKR) coefficient as

$$s_{ij} = \begin{cases} \frac{\kappa_{ij}}{\sum_{\mathbf{x}_{j'} \in \mathcal{N}_i} \kappa_{ij'}}, & \text{if } \mathbf{x}_j \in \mathcal{N}_i, \\ 0, & \text{if } \mathbf{x}_j \notin \mathcal{N}_i. \end{cases} \quad (29)$$

From the perspective of single kernel structure exploration, we highlight the differences between these MKC methods and the proposed LKR model in terms of localized reconstruction, optimal neighbor weight assignment, and enhanced local weight determination.

In contrast to performing global reconstruction with all remaining samples, as done by KSC and LKSC, the LKR model confines its reconstruction efforts to the local clique. This approach presents two key advantages. Firstly, it significantly enhances the ability of LKR model to explore the local manifold structure, particularly beneficial

Table 2
Description of the data sets.

	Name	#Instances	#Features	#Classes
D1	Trachea	1013	13741	7
D2	BA	1404	320	36
D3	COIL	1440	1024	20
D4	MFEAT	2000	240	10
D5	HiTech	2301	22498	6
D6	K1B	2340	21839	6
D7	LI	3362	16418	15
D8	Fat	3618	15492	9
D9	MNIST	4000	784	10
D10	MACOSKO	6418	8608	39

for nonlinear clustering. Secondly, it results in a notable improvement in computational efficiency, as the neighborhood size k is much smaller than the total number of samples n .

While KLSL and KALSLS primarily focus on maximizing pairwise alignment and the entropy of uniform weight to explore the local structure, the proposed LKR model takes a step further. Derived from the local reconstruction criteria in Eq (4), the LKR model minimizes an additional term: $\sum_{u=1}^k \sum_{v=1}^k \bar{a}_{uv}^i \bar{a}_{uv}^i$. This innovative addition reduces redundancy and enriches the diversity of neighbor coefficients. As a result, the superior performance of LKR model, as demonstrated in the ablation study in Section 5.5, can be attributed to this unique part. Notably, this concept of optimal neighbor weight assignment represents a previously unexplored facet within existing MKC methods.

In contrast to methods such as NK- k , NK- τ , and LNWKRC, where local weight determination is primarily influenced by the similarity in the original kernel, i.e., κ_{ij} , the LKR model takes a more comprehensive approach. It considers not only the original similarity but also the relationships among neighbors within the same clique. This approach results in a more nuanced determination of local weights.

5. Experiments

In this section, we perform comprehensive experiments on benchmark datasets to evaluate the proposed algorithm in terms of clustering performance and running time, comparing it with recently developed methods for MKC. Additionally, we conduct meticulous ablation studies that focus on two aspects: the exploration of local structure using a single kernel and the exploitation of candidate graph enhancement and integration. Furthermore, we provide a parameter sensitivity analysis to assess the robustness of the proposed algorithm.

5.1. Data sets and evaluation metric

We employed a diverse range of datasets from multiple domains, exhibiting significant variations in sample sizes, dimensionalities, class balances, and sparsity. We simply introduce these datasets as follows. The BA dataset¹ contains images of letters and digits with pixel features. The COIL dataset² comprises images of objects described by pixel features. The MFEAT dataset³ includes multiple features used to describe images of letters and digits. The MNIST dataset⁴ is a well-known collection of handwritten digit images. The HiTech and K1B datasets are text corpora for text clustering in CLUTO [47]. The Trachea, Large Intestine (LI), and Fat datasets are three single-cell RNA-seq datasets sourced from the Tabula Muris project,⁵ aiming to create a comprehensive single-cell transcriptomic atlas of major organs and tissues in

¹ <https://cs.nyu.edu/~roweis/data/binaryalphadigs.mat>

² <http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php>

³ <https://archive.ics.uci.edu/ml/datasets/Multiple+Features>

⁴ <http://www.cad.zju.edu.cn/home/dengcai/Data/MNIST/2k2k.mat>

⁵ https://figshare.com/articles/dataset/Single-cell_RNA-seq_data_from_Smart-seq2_sequencing_of_FACS_sorted_cells/5715040

laboratory mice. The Macosko⁶ dataset refers to the single-cell RNA sequencing data obtained through the Drop-seq method [48], providing highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. The statistical characteristics of the datasets are summarized in Table 2. To ensure reproducible experiments, we have made all these datasets available on GitHub.⁷

Four clustering evaluation measures, including Accuracy (ACC), Normalized Mutual Information (NMI), Rand Index (RI), and F1 score, are adopted to assess the quality of clustering results compared to the ground truth. ACC measures the percentage of correctly assigned data points, NMI quantifies the similarity between the clustering result and the ground truth, and RI measures the agreement between two partitions. The F1 score combines precision and recall to provide a balanced evaluation of clustering accuracy. These four metrics yield values ranging from 0 to 1, where higher values indicate a stronger agreement between the clustering result and the ground truth.

5.2. Comparison methods

To assess the effectiveness of our proposed algorithm, we compared it with ten state-of-the-art MKC algorithms. These algorithms include six MKC-CKL methods, i.e., ONALK [11], MKCSS [1], SPMKC [8], LSWMKC [6], SMKMM [27], and LSMKMM [12], one MKC-NCKL method, i.e., CAGL [5], and three MKC-LF methods, i.e., PMKSC [32], LF-PGR [33], and LFLKA [4]. Similar to CAGL, our method can be categorized into the type of MKC-NCKL. It is worthwhile to mention that all these compared methods, except for SMKMM, also consider the local structure.

5.3. Experiment setup

In line with the strategies employed in previous studies such as [5, 13, 23], we extract 12 base kernels using various configurations of kernel functions and parameters to investigate the clustering performance of MKC approaches. Specifically, we consider seven Gaussian kernels with different bandwidths, denoted as $\kappa_{ij} = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\rho\delta^2}\right)$, where ρ takes values in the range of [0.01, 0.05, 0.1, 1, 10, 50, 100], and δ corresponds to the maximum distance between any two samples. Additionally, we include four polynomial kernels, represented as $\kappa_{ij} = (a + \mathbf{x}_i^T \mathbf{x}_j)^b$, where a varies in the range of [0, 1], and b takes values in [2, 4]. Lastly, we include one linear kernel, denoted as $\kappa_{ij} = \mathbf{x}_i^T \mathbf{x}_j$. To ensure consistency, all candidate kernels undergo centralization and normalization to achieve a unit trace.

We set the number of clusters for all these methods to the true number of classes. As most MKC methods involve parameter tuning, we employ the commonly used grid search strategy [5] to ensure a fair comparison. For each clustering algorithm, we run it with the parameters suggested in their original papers and report the best results obtained. The two parameters involved in LKRGDF are tuned by restricting $k \in [5, 10, 15, 20]$ and $\eta \in [3, 5, 7, 9]$. To obtain reliable results, we compute the low-dimensional embeddings from all the compared methods and report the averaged results of ten independent runs of the k-means algorithm. Within each run, we repeat the k-means process 100 times and record the clustering result with the minimal objective value. The running time of each algorithm is also reported. All experiments are conducted on a workstation equipped with an Intel Core i7 8700K CPU (3.7 GHz), 64-GB RAM, and MATLAB 2020b (64-bit). To ensure reproducibility, we utilize the original code provided by the authors for all the baselines. Additionally, we have made the code for our proposed method available at <https://github.com/YanChenSCU/LKRGDF-2023.git>.

5.4. Clustering result analysis

Visualizing Clustering Results. To provide a more intuitive visualization of the performance of the compared algorithms, we utilize t-distributed Stochastic Neighbor Embedding (t-SNE) [49] to visualize the obtained cluster structures on the MNIST dataset. The perplexity for t-SNE in all these methods is set to the default value of 30. As depicted in Fig. 3, it is evident that LKGRDF shows clear cluster structures, thereby validating its superior performance compared to the other algorithms. This visual confirmation further strengthens the evidence of the enhanced clustering capability of our method.

Quantitative Clustering Results. We present a detailed analysis of the clustering results, including four evaluation measures for the 11 MKC methods across ten benchmark datasets. The results are summarized in Table 3, and our analysis reveals two key observations. First, the early fusion methods consistently outperform their late fusion counterparts. This trend can be attributed to the fact that late fusion approaches heavily rely on the top eigenvectors of candidate kernels, which may not effectively capture local structures compared to methods that explore the local characteristics of the kernel. This highlights the importance of leveraging high-quality inputs with clear cluster structures through diverse exploration techniques. Second, our proposed method consistently demonstrates top-tier performance, either securing the best or second-best results across various datasets. For instance, on the D9 dataset, the clustering accuracy of the second-best algorithm (CAGL) is 66.75% in terms of ACC, while our method achieves a significantly higher accuracy of 74.50%. When considering the average results across all ten datasets, our method attains an ACC of 77.16%, surpassing LSWMKC (the best among MKC-CKL methods) with an improvement of 6.49%, CAGL (the best among MKC-NCKL methods) with an improvement of 10.75%, and LSMKMM (the best result among MKC-LF methods) with an improvement of 12.38%. These consistent enhancements in various evaluation metrics underscore the effectiveness of our proposed approach compared to existing methods.

Computation Time Comparison. In addition to evaluating the clustering performance, we also recorded the CPU time of the algorithms, as shown in Table 4. The values in each cell, labeled D1 to D10, represent the total tuning time of the method across all hyperparameters. The average tuning time is also provided below. The grid row corresponds to the number of hyperparameter combinations, while the last row displays the average running time obtained by dividing the average tuning time by the number of grids.

Based on these results in Tables 3 and 4, we have three aspects of observations. First, in comparison to methods that involve local structure learning, such as MKCSS, SPMKC, CAGL, ONALK, PMKSC, LSMKMM, and LSWMKC, our method demonstrates superior performance with fewer parameters to tune and shorter overall time consumption. The average tuning time for our method is 1470 s, which is less than 1/5 of the time consumed by the second-best method, LSWMKC. Second, in contrast to late fusion methods, our approach demands more time on benchmark datasets. This increase in computational cost can be attributed to the meticulous exploration and utilization of local manifold structures within the kernel space. However, this moderate increase in computational cost is justified by the substantial enhancement in clustering performance. Third, when considering a comparison with the second-best algorithm, LSWMKC, our method achieves a 6.49% improvement in clustering accuracy while requiring less than 1/5 of the time consumption, as demonstrated in Tables 3 and 4. These results clearly demonstrate the superiority of our method in terms of both effectiveness and efficiency. Therefore, our method is highly suitable for the task of MKC.

⁶ <https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE63473>

⁷ <https://github.com/YanChenSCU/LKRGDF-2023.git>

Table 3

Clustering Results in terms of ACC, NMI, RI, and F1 of 11 compared methods on ten benchmark data sets. The best results are in bold, and the second-best results are underlined.

	MKCSS	SPMKC	CAGL	ONALK	PMKSC	LFPGR	LFLKA	LSMKKM	LSWMKC	SMKKM	LKRGDF
ACC(%)											
D1	82.03	<u>83.81</u>	83.61	72.66	61.05	58.36	62.87	63.53	80.55	54.31	89.83
D2	<u>52.14</u>	45.08	46.30	48.02	47.64	42.94	46.66	45.95	51.74	42.59	55.32
D3	78.26	84.58	88.12	70.04	77.07	67.22	68.63	65.67	<u>98.75</u>	61.28	99.10
D4	96.80	80.60	96.40	85.35	75.10	74.93	76.89	80.82	96.90	76.92	96.85
D5	<u>46.06</u>	41.50	43.02	45.98	38.82	38.67	44.68	45.24	45.98	38.61	49.39
D6	85.94	82.91	92.52	82.99	87.09	84.19	85.09	80.90	90.00	81.36	<u>91.58</u>
D7	43.25	<u>44.69</u>	44.08	40.80	41.53	43.31	40.51	42.15	43.35	38.78	51.79
D8	79.10	74.52	79.24	57.64	59.04	55.53	55.93	53.37	<u>81.72</u>	51.59	84.15
D9	65.30	60.32	<u>66.75</u>	56.95	55.97	54.61	56.53	55.99	64.30	54.66	74.50
D10	63.56	65.21	56.65	63.47	66.09	62.44	65.44	61.93	<u>71.32</u>	62.91	79.15
Avg	69.24	66.32	69.67	62.39	60.94	58.22	60.32	59.56	<u>72.46</u>	56.30	77.16
NMI(%)											
D1	76.64	78.77	<u>79.67</u>	60.70	59.33	58.17	58.97	58.11	76.86	53.04	81.17
D2	65.38	61.49	60.66	61.12	61.14	58.05	60.55	59.78	<u>65.73</u>	57.60	66.75
D3	88.44	92.32	94.50	78.40	85.49	77.62	77.28	76.04	<u>98.73</u>	74.11	98.97
D4	<u>92.96</u>	83.35	92.35	76.63	72.55	67.95	71.59	73.26	93.22	70.65	92.68
D5	28.00	23.02	19.81	25.95	19.96	23.57	24.17	23.67	26.44	18.19	27.90
D6	74.73	56.87	<u>81.10</u>	58.82	71.04	66.09	69.77	59.44	77.43	51.34	81.25
D7	52.20	<u>55.05</u>	<u>50.56</u>	48.64	49.66	49.71	48.51	48.58	52.30	48.55	56.17
D8	70.74	78.70	77.77	54.09	54.31	52.03	50.46	52.44	77.33	48.87	<u>78.01</u>
D9	68.95	63.08	<u>70.48</u>	50.76	50.38	48.84	51.61	51.47	66.51	49.94	72.72
D10	72.27	<u>77.02</u>	<u>62.24</u>	75.27	75.98	72.82	75.81	73.91	76.54	74.70	79.67
Avg	69.03	66.97	68.91	59.04	59.98	57.48	58.87	57.67	<u>71.11</u>	54.70	73.53
RI(%)											
D1	71.19	73.30	<u>75.13</u>	57.71	46.73	42.79	41.88	49.52	70.65	35.20	76.95
D2	37.27	32.71	29.58	32.66	33.45	28.54	31.36	30.23	<u>37.38</u>	27.73	39.52
D3	75.81	80.74	84.60	62.90	72.29	61.28	60.19	57.94	<u>97.54</u>	53.84	98.20
D4	93.06	75.37	92.13	71.55	64.14	59.22	62.41	65.75	93.18	62.14	<u>93.12</u>
D5	<u>21.51</u>	14.66	15.39	18.99	11.71	17.56	17.63	18.84	21.12	11.10	22.12
D6	77.85	60.07	90.17	65.07	74.21	68.17	72.83	62.03	83.14	58.24	89.57
D7	31.51	34.57	<u>36.12</u>	28.05	27.99	28.22	25.93	29.13	31.91	26.67	38.64
D8	69.10	69.05	76.80	44.63	46.44	39.95	38.71	38.26	74.21	34.27	75.80
D9	<u>57.31</u>	49.73	57.00	38.68	37.94	37.38	38.32	37.82	55.00	36.27	64.13
D10	61.49	<u>67.33</u>	39.31	51.65	56.24	49.68	56.79	54.13	62.35	53.61	71.99
Avg	59.61	55.75	59.62	47.19	47.11	43.28	44.61	44.36	<u>62.65</u>	39.91	67.00
F1(%)											
D1	76.45	78.17	<u>79.62</u>	64.89	55.25	52.19	51.48	57.62	76.03	46.33	81.10
D2	39.06	34.86	32.05	34.57	35.37	30.53	33.30	32.28	<u>39.19</u>	29.81	41.25
D3	77.08	81.75	85.46	64.76	73.74	63.22	62.20	60.10	<u>97.66</u>	56.25	98.29
D4	93.75	78.01	92.92	74.42	67.76	63.31	66.22	69.21	93.86	65.98	<u>93.81</u>
D5	38.46	38.81	<u>40.92</u>	39.17	34.79	34.48	36.74	39.20	37.83	35.72	41.16
D6	86.24	79.09	94.13	79.63	83.63	79.13	83.43	77.65	89.76	77.41	93.70
D7	38.03	41.33	44.37	34.05	34.29	34.48	31.89	35.24	38.90	33.32	<u>43.99</u>
D8	73.45	74.27	80.97	52.48	53.70	48.03	47.04	46.46	78.10	43.19	<u>79.40</u>
D9	62.03	55.43	<u>62.14</u>	45.03	44.41	43.80	44.77	44.36	59.94	43.05	67.96
D10	63.32	<u>69.02</u>	<u>44.31</u>	53.49	58.09	51.64	58.64	56.18	64.33	55.61	73.65
Avg	64.79	63.07	65.69	54.25	54.10	50.08	51.57	51.83	<u>67.56</u>	48.67	71.43

Table 4

Execution time of compared algorithms.

	MKCSS	SPMKC	CAGL	ONALK	PMKSC	LFPGR	LFLKA	LSMKKM	LSWMKC	SMKKM	LKRGDF
D1	384	383	2667	6815	18773	20	69	56	158	13	60
D2	1139	3041	7183	10987	54510	229	215	257	403	128	131
D3	1310	1450	5479	17805	34392	220	206	150	454	25	185
D4	3033	4098	13256	35432	87615	121	433	730	1317	138	378
D5	4490	7488	72845	39638	92578	718	673	1412	2229	108	468
D6	6316	8678	90666	47080	98449	1653	699	888	2264	154	480
D7	15923	28611	70298	205597	365425	398	2540	1514	6690	158	1751
D8	21575	41102	90856	239166	317001	439	2870	2032	8362	194	1921
D9	20175	35358	108735	254367	306472	782	3435	3458	11804	580	1980
D10	93518	211276	460714	848244	731046	11498	15880	20800	46406	2920	7346
Avg Tune	16786	34148	92270	170513	210626	1608	2702	3130	8009	442	1470
Grids	16	36	162	341	216	25	247	19	11	1	16
Avg Run	1049	949	570	500	975	64	11	165	728	442	92

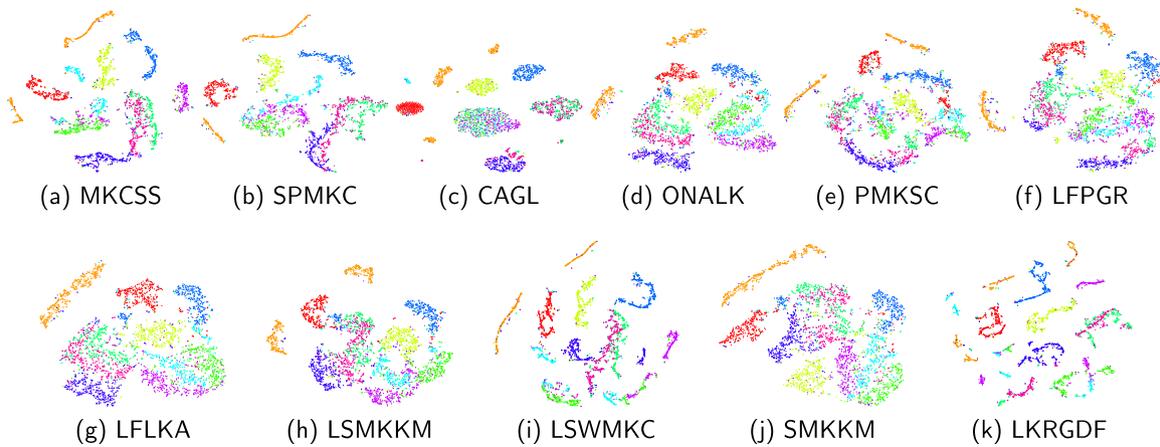


Fig. 3. Visualization of the revealed cluster structure of the compared algorithms with t-SNE [49] on MNIST.

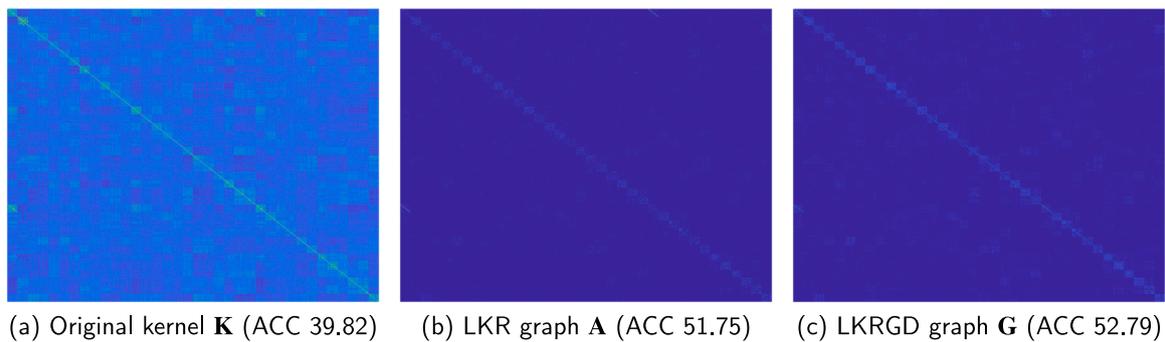


Fig. 4. Visualization of original kernel \mathbf{K} , LKR graph \mathbf{A} and LKRGD graph \mathbf{G} , with their result in terms of ACC(%).

5.5. Ablation study

Comparing LKR with Existing Strategies. In the subsequent, we conduct an ablation study to highlight the improvements achieved through various kernel structure exploration strategies. Our comprehensive comparison encompasses nine methods, including the baseline method KKM, as well as seven existing local clustering strategies: KSC, NK- k , NK- τ , KLSL, KALSL, LKSC, and LNWKR, as discussed in Section 4.5. Additionally, we incorporate our proposed LKR model, detailed in Section 3.1. Each of these methods is independently applied to 12 distinct kernels. For the local methods, we extract an affinity graph from each kernel and employ spectral clustering to derive a single result. The results of this analysis are presented in Table 5, offering a summary of the average performance across the 12 kernels, utilizing four evaluation measures across ten benchmark datasets.

The results lead to three key observations. Firstly, all eight strategies consistently outperform the standard KKM, demonstrating their effectiveness in identifying more meaningful cluster structures and emphasizing the importance of improved single kernel structure extraction. Secondly, the average results of the single kernel methods (Table 5) are competitive with those of the MKC methods (Table 3). This challenges the conventional belief that MKC methods should significantly outperform their single kernel counterparts, highlighting the need for a robust single kernel structure extraction strategy. Lastly, the average results of LKR on 12 kernels surpass not only other single-kernel cluster extraction strategies but also most of the compared MKC methods. The substantial improvements achieved by LKR can be primarily attributed to the introduction of a new term that minimizes redundancy among local neighbors. These findings underscore the importance of carefully designing a strong single kernel structure extraction strategy for the successful application of MKC methods. The

effectiveness of LKR highlights its advantages in capturing meaningful cluster structures.

Visualizing Affinity Graphs in LKR and LKRGD. In this illustration, using a polynomial kernel from the D2 dataset, we visually distinguish the original kernel \mathbf{K} , the affinity graph \mathbf{A} generated by the LKR model, and the enhanced affinity graph \mathbf{G} achieved through the graph diffusion procedure, denoted as the LKRGD model. The accompanying figures and their associated clustering accuracy are presented in Fig. 4.

Evidently, the clustering accuracy exhibits a gradual improvement, transitioning from the original kernel (ACC: 39.82%) to the LKR graph (ACC: 51.75%) and finally to the LKRGD graph (ACC: 52.79%). Our observations highlight the impact of the density of original kernel, which leads to less satisfactory clustering performance. The LKR method effectively captures the local affinity structure, leading to an enhancement in the clustering results. Finally, the LKRGD model aggregates graphs of different orders, harnessing both first-order and long-range connections to further refine the clustering analysis. This qualitative evaluation, particularly in terms of ACC, reinforces our observations, emphasizing the incremental improvement in clustering performance throughout this process.

Comparing LKR, LKRGD, and LKRGDF. In the subsequent analysis, we conduct a comprehensive exploration of the synergistic effects of three critical components of our method: LKR in Section 3.1, GD in Section 3.2, and MGF in Section 3.3. We offer a detailed analysis of the results obtained from these three consecutive stages, with each stage building upon the previous one. To be specific, we investigate the LKR model applied to affinity graphs $\{\mathbf{A}^r\}_{r=1}^m$ as a strong baseline, followed by the LKR model enhanced by GD, using affinity graphs $\{\mathbf{G}^r\}_{r=1}^m$, and culminating in the comprehensive LKR with Graph Diffusion and Fusion (LKRGDF) on the consensus affinity matrix \mathbf{S} . The results from these three key steps are presented in the last three columns of Table 5.

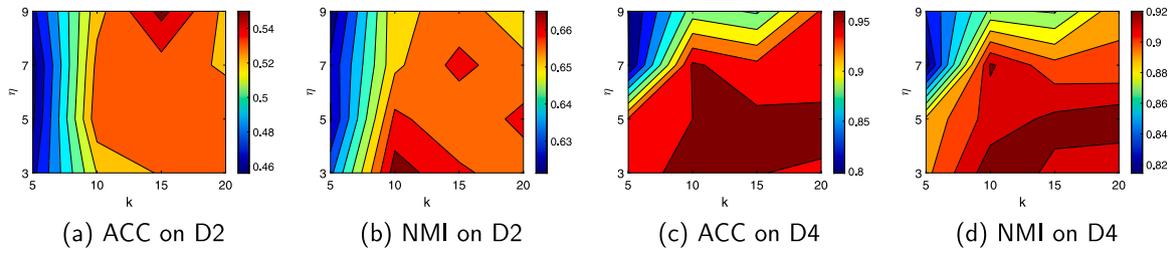


Fig. 5. Parameter sensitivity study of neighborhood size k and decay parameter η on D2 and D4.

Table 5

ACC, NMI, RI, and F1 results of different single kernel cluster structure extraction method on the benchmark datasets.

	KKM	KSC	NK- k	NK- τ	KLSL	KALSLS	LKSC	LNWKR	LKR	LKRGD	LKRGDF	
D1	ACC	57.11	73.28	78.13	64.48	75.61	74.23	82.21	68.81	81.28	84.05	89.83
	NMI	53.80	70.86	73.72	56.43	73.30	66.76	77.23	57.90	77.99	78.72	81.17
	RI	41.77	64.30	69.05	46.10	64.21	58.39	72.12	49.09	72.61	72.51	76.95
	F1	51.19	70.72	74.80	57.86	70.76	66.93	77.12	60.43	77.53	77.54	81.10
D2	ACC	40.20	45.84	51.76	46.67	49.39	44.95	45.11	47.03	53.21	53.15	53.32
	NMI	55.43	59.72	64.56	60.38	63.95	55.63	60.52	59.48	66.77	65.42	66.75
	RI	25.39	30.30	36.99	31.08	35.60	18.59	32.03	26.20	39.67	38.17	39.52
	F1	27.46	32.23	38.82	32.99	37.47	21.90	34.10	28.61	41.39	39.89	41.25
D3	ACC	63.57	67.98	82.83	67.39	75.67	84.36	84.30	82.81	90.84	84.96	99.10
	NMI	75.36	77.65	89.05	77.30	89.46	92.33	93.57	90.70	96.01	94.09	98.97
	RI	56.45	60.83	78.28	60.50	75.12	79.31	82.91	77.09	89.12	82.91	98.20
	F1	58.67	62.84	79.39	62.50	76.53	80.50	83.85	78.38	89.69	83.85	98.29
D4	ACC	71.49	75.07	95.21	81.58	83.96	93.96	75.72	85.52	96.02	97.40	96.85
	NMI	67.11	71.18	91.25	81.79	85.44	90.91	82.38	88.04	91.14	93.93	92.68
	RI	58.15	62.56	90.81	75.96	79.92	89.74	73.83	82.77	91.36	94.29	93.12
	F1	62.38	66.36	91.74	78.52	82.14	90.80	76.67	84.62	92.22	94.86	93.81
D5	ACC	38.24	41.22	40.40	42.02	48.67	36.49	39.72	42.14	43.46	48.78	49.39
	NMI	21.63	24.06	22.39	25.65	27.53	11.50	22.39	20.35	25.33	27.94	27.90
	RI	14.48	17.87	17.06	18.52	23.52	7.56	15.93	18.41	19.57	23.67	22.12
	F1	31.94	35.43	35.18	34.70	39.19	36.62	34.60	41.93	37.94	39.32	41.16
D6	ACC	64.31	79.30	73.73	80.99	89.68	84.61	77.75	83.08	90.38	91.62	91.58
	NMI	48.56	65.83	56.23	65.63	74.02	56.24	57.54	66.71	79.72	83.18	81.25
	RI	37.77	65.95	52.73	64.31	81.91	62.83	56.05	70.44	84.31	91.76	89.57
	F1	56.87	77.82	70.15	77.43	89.53	81.16	74.77	83.45	90.79	95.12	93.70
D7	ACC	40.45	42.95	42.52	38.17	44.34	33.87	44.18	32.06	45.26	44.20	51.79
	NMI	48.33	51.20	51.90	45.79	52.78	37.97	54.76	34.93	53.36	53.87	56.17
	RI	26.35	32.00	30.53	23.67	32.33	19.61	34.91	18.63	33.20	32.98	38.64
	F1	32.77	38.85	37.01	30.88	38.50	30.68	41.52	29.11	39.70	39.45	43.99
D8	ACC	56.05	69.52	75.87	55.73	75.91	70.26	72.31	56.82	80.54	85.81	84.15
	NMI	51.39	65.52	72.89	50.73	71.94	64.64	76.71	52.87	74.78	78.03	78.01
	RI	40.01	55.73	69.42	32.30	66.54	58.38	67.46	40.47	71.34	78.19	75.80
	F1	48.16	62.35	73.88	43.08	71.64	66.16	72.89	52.86	75.64	81.47	79.40
D9	ACC	53.64	55.13	69.24	65.57	63.47	63.54	58.89	71.71	65.23	71.68	74.50
	NMI	46.82	52.90	68.69	61.99	65.15	67.14	61.49	69.34	66.83	70.74	72.72
	RI	34.41	40.20	57.23	50.84	53.22	55.18	47.70	59.30	54.14	60.70	64.13
	F1	41.10	46.53	61.88	56.22	58.41	60.58	53.58	63.79	59.27	64.91	67.96
D10	ACC	61.28	61.41	56.72	48.65	57.08	65.24	63.93	55.96	63.86	69.77	79.15
	NMI	74.39	74.33	67.89	63.87	68.34	64.04	75.42	67.14	72.01	75.72	79.67
	RI	50.17	50.32	48.34	42.30	46.15	39.99	54.92	40.17	53.10	60.62	71.99
	F1	52.06	52.24	50.57	44.50	48.63	44.95	56.84	43.18	55.30	62.67	73.65
Avg	ACC	54.63	61.17	66.64	59.12	66.38	65.15	64.41	62.59	71.01	73.14	77.16
	NMI	54.28	61.33	65.86	58.96	67.19	60.72	66.20	60.75	70.39	72.16	73.53
	RI	38.49	48.01	55.05	44.56	55.85	48.96	53.79	48.26	60.84	63.58	67.00
	F1	46.26	54.54	61.34	51.87	61.28	58.03	60.60	56.64	65.95	67.91	71.43

Upon thorough analysis of these results, two key observations emerge. Firstly, it becomes evident that the global diffusion process augments the performance of individual kernels. This enhancement underscores the critical role of global diffusion in accurately identifying long-range connections within the data. Secondly, the effectiveness of our MKC method, LKRGDF, becomes manifest when comparing the

LKRGD results obtained from individual kernels. The noticeable improvement achieved through the MGF procedure provides compelling evidence for the efficacy of our proposed approach. These combined findings from our quantitative and qualitative studies serve as robust validation of the effectiveness of the essential procedural steps in the MKC task: LKR, GD, and MGF.

5.6. Parameters sensitivity study

Our method involves tuning two parameters: k and η . The parameter k is used to control the neighbor sparsity during the LKR stage and fusion process, while the parameter η determines the rate of eigenvalue exponential decay in the diffusion step. The two parameters are searched by restricting $k \in [5, 10, 15, 20]$ and $\eta \in [3, 5, 7, 9]$. Fig. 5 showcases the sensitivity of clustering performance to variations in k and η on two different datasets. The results depicted in the figure highlight the robustness of our method across diverse datasets, indicating its stability under a wide range of parameter combinations.

6. Conclusion

This paper unveils LKRGDF, a novel approach to MKC. Moving away from traditional local MKC methods that predominantly depend on pairwise similarity for affinity, our method utilizes LKR coefficients to effectively capture local affinity. This approach mitigates redundancy and enhances the discrimination of highly similar neighborhood samples. Furthermore, we seamlessly incorporate these affinities using graph heat kernel diffusion. Leveraging the low-pass filter property of global diffusion, we can smooth high-order connections and focus on the most salient clusters. Rather than applying a unified learning framework, we implement a sequential pipeline comprising three pivotal steps: LKR, GD, and MGF. This structure enables us to proficiently explore and exploit local structures within kernels. Experimental outcomes derived from ten benchmark datasets unequivocally showcase the superior performance and computational efficiency of our method when compared to ten recently developed alternatives. In the future, we will consider how to perform LKR and consensus graph learning jointly efficiently.

CRedit authorship contribution statement

Yan Chen: Conceptualization, Collected and analyzed the data, Writing – original draft. **Liang Du:** Provided critical feedback, Assisted in the data analysis, Revision of the manuscript. **Peng Zhou:** Conducted experiments, Performed simulations, Data interpretation. **Lei Duan:** Study design, provided expertise in multiple source fusion techniques, Assisted in the manuscript preparation. **Yuhua Qian:** Theoretical framework, Performed statistical analysis, Revised the manuscript.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The code and data has been published on github for reproducible experiments.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (61972268, 62376146, 61976129, 62176001, 62136005), in part by the Natural Science Project of Anhui Provincial Education Department grants 2023AH030004, in part by the Special Fund Project for Guiding Local Science and Technology Development by the Central Government (No.: YDZJSX20231D003) and GHfundx A (202302017223).

References

- [1] S. Zhou, X. Liu, M. Li, E. Zhu, L. Liu, C. Zhang, J. Yin, Multiple kernel clustering with neighbor-kernel subspace segmentation, *IEEE Trans. Neural Netw. Learn. Syst.* 31 (4) (2020) 1351–1362.
- [2] Y. Yang, D. Xu, F. Nie, S. Yan, Y. Zhuang, Image clustering using local discriminant models and global integration, *IEEE Trans. Image Process.* 19 (10) (2010) 2761–2773.
- [3] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, *Science* 290 (5500) (2000) 2323–2326.
- [4] T. Zhang, X. Liu, L. Gong, S. Wang, X. Niu, L. Shen, Late fusion multiple kernel clustering with local kernel alignment maximization, *IEEE Trans. Multimed.* 25 (2023) 993–1007.
- [5] Z. Ren, S.X. Yang, Q. Sun, T. Wang, Consensus affinity graph learning for multiple kernel clustering, *IEEE Trans. Cybern.* 51 (6) (2021) 3273–3284.
- [6] L. Li, S. Wang, X. Liu, E. Zhu, L. Shen, K. Li, K. Li, Local sample-weighted multiple kernel clustering with consensus discriminative graph, *IEEE Trans. Neural Netw. Learn. Syst.* (2024) 1–14, In Press.
- [7] Z. Ren, H. Li, C. Yang, Q. Sun, Multiple kernel subspace clustering with local structural graph and low-rank consensus kernel learning, *Knowl.-Based Syst.* (ISSN: 0950-7051) 188 (2020) 105040.
- [8] Z. Ren, Q. Sun, Simultaneous global and local graph structure preserving for multiple kernel clustering, *IEEE Trans. Neural Netw. Learn. Syst.* 32 (5) (2021) 1839–1851.
- [9] M. Li, X. Liu, L. Wang, Y. Dou, J. Yin, E. Zhu, Multiple kernel clustering with local kernel alignment maximization, in: *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, 2016, pp. 1704–1710.
- [10] X. Zhu, X. Liu, M. Li, E. Zhu, L. Liu, Z. Cai, J. Yin, W. Gao, Localized incomplete multiple kernel k-means, in: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, 2018, pp. 3271–3277.
- [11] J. Liu, X. Liu, J. Xiong, Q. Liao, S. Zhou, S. Wang, Y. Yang, Optimal neighborhood multiple kernel clustering with adaptive local kernels, *IEEE Trans. Knowl. Data Eng.* 34 (6) (2022) 2872–2885.
- [12] X. Liu, S. Zhou, L. Liu, C. Tang, S. Wang, J. Liu, Y. Zhang, Localized simple multiple kernel K-means, in: *Proceedings of 2021 IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9293–9301.
- [13] L. Du, P. Zhou, L. Shi, H. Wang, M. Fan, W. Wang, Y.-D. Shen, Robust multiple kernel k-means using l21-norm, in: *Proceedings of Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015, pp. 3476–3482.
- [14] X. Liu, S. Zhou, Y. Wang, M. Li, Y. Dou, E. Zhu, J. Yin, Optimal neighborhood kernel clustering with multiple kernels, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017, pp. 2266–2272.
- [15] H.-C. Huang, Y.-Y. Chuang, C.-S. Chen, Multiple kernel fuzzy clustering, *IEEE Trans. Fuzzy Syst.* 20 (1) (2012) 120–134.
- [16] M. Gönen, A.A. Margolin, Localized data fusion for kernel k-means clustering with application to cancer biology, *Adv. Neural Inf. Process. Syst.* 27 (2014) 1305–1313.
- [17] X. Liu, Y. Dou, J. Yin, L. Wang, E. Zhu, Multiple kernel k-means clustering with matrix-induced regularization, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, 2016, pp. 1888–1894.
- [18] Z. Kang, C. Peng, Q. Cheng, Twin learning for similarity and clustering: A unified kernel approach, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017, pp. 2080–2086.
- [19] Z. Kang, C. Peng, Q. Cheng, Z. Xu, Unified spectral clustering with optimal graph, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018, pp. 3366–3373.
- [20] Z. Kang, C. Peng, Q. Cheng, X. Liu, X. Peng, Z. Xu, L. Tian, Structured graph learning for clustering and semi-supervised classification, *Pattern Recognit.* 110 (2021) 107627.
- [21] S. Zhou, E. Zhu, X. Liu, T. Zheng, Q. Liu, J. Xia, J. Yin, Subspace segmentation-based robust multiple kernel clustering, *Inf. Fusion* 53 (2020) 145–154.
- [22] K. Zhao, L. Wen, W. Chen, Z. Xu, Low-rank kernel learning for graph-based clustering, *Knowl.-Based Syst.* 163 (JAN.1) (2019) 510–517.
- [23] C. Yang, Z. Ren, Q. Sun, M. Wu, M. Yin, Y. Sun, Joint correntropy metric weighting and block diagonal regularizer for robust multiple kernel subspace clustering, *Inform. Sci.* 500 (2019) 48–66.
- [24] Z. Kang, X. Lu, J. Yi, Z. Xu, Self-weighted multiple kernel learning for graph-based clustering and semi-supervised classification, in: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, 2018, pp. 2312–2318.
- [25] Z. Ren, H. Lei, Q. Sun, C. Yang, Simultaneous learning coefficient matrix and affinity graph for multiple kernel clustering, *Inform. Sci.* 547 (2021) 289–306.
- [26] Y. Lu, L. Wang, J. Lu, J. Yang, C. Shen, Multiple kernel clustering based on centered kernel alignment, *Pattern Recognit.* 47 (11) (2014) 3656–3664.
- [27] X. Liu, SimpleMKM: Simple multiple kernel K-means, *IEEE Trans. Pattern Anal. Mach. Intell.* 45 (4) (2023) 5174–5186.
- [28] X. Li, Z. Ren, H. Lei, Y. Huang, Q. Sun, Multiple kernel clustering with pure graph learning scheme, *Neurocomputing* 424 (2021) 215–225.
- [29] Z. Ren, M. Mukherjee, J. Lloret, P. Venu, Multiple kernel driven clustering with locally consistent and selfish graph in industrial IoT, *IEEE Trans. Ind. Inform.* 17 (4) (2021) 2956–2963.
- [30] Z. Ren, Q. Sun, D. Wei, Multiple kernel clustering with kernel k-means coupled graph tensor learning, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35, no. 11, 2021, pp. 9411–9418.
- [31] S. Wang, X. Liu, E. Zhu, C. Tang, J. Liu, J. Hu, J. Xia, J. Yin, Multi-view clustering via late fusion alignment maximization, in: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, International Joint Conferences on Artificial Intelligence Organization*, 2019, pp. 3778–3784.
- [32] M. Sun, S. Wang, P. Zhang, X. Liu, X. Guo, S. Zhou, E. Zhu, Projective multiple kernel subspace clustering, *IEEE Trans. Multimed.* 24 (2022) 2567–2579.
- [33] S. Wang, X. Liu, L. Liu, S. Zhou, E. Zhu, Late fusion multiple kernel clustering with proxy graph refinement, *IEEE Trans. Neural Netw. Learn. Syst.* 34 (8) (2023) 4359–4370.
- [34] L.K. Saul, S.T. Roweis, Think globally, fit locally: Unsupervised learning of low dimensional manifolds, *J. Mach. Learn. Res.* 4 (Jun) (2003) 119–155.
- [35] F. Chung, The heat kernel as the pagerank of a graph, *Proc. Natl. Acad. Sci.* 104 (50) (2007) 19735–19740.
- [36] Z. Wu, S. Pan, G. Long, J. Jiang, C. Zhang, Beyond low-pass filtering: Graph convolutional networks with automatic filtering, *IEEE Trans. Knowl. Data Eng.* 35 (7) (2023) 6687–6697.
- [37] F. Zhang, E.R. Hancock, Graph spectral image smoothing using the heat kernel, *Pattern Recognit.* 41 (11) (2008) 3328–3342.
- [38] B. Hu, X. Wang, P. Zhou, L. Du, Multi-view outlier detection via graphs denoising, *Information Fusion (ISSN: 1566-2535)* 101 (2024) 102012.
- [39] P. Zhou, L. Du, X. Li, Adaptive consensus clustering for multiple k-means via base results refining, *IEEE Trans. Knowl. Data Eng.* 35 (10) (2023) 10251–10264.
- [40] H.-T. Wai, S. Segarra, A.E. Ozdaglar, A. Scaglione, A. Jadbabaie, Blind community detection from low-rank excitations of a graph filter, *IEEE Trans. Signal Process.* 68 (2020) 436–451.
- [41] F. Nie, X. Wang, H. Huang, Clustering and projected clustering with adaptive neighbors, in: *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2014, pp. 977–986.
- [42] P. Zhou, L. Du, X. Liu, Y.-D. Shen, M. Fan, X. Li, Self-paced clustering ensemble, *IEEE Trans. Neural Netw. Learn. Syst.* 32 (4) (2021) 1497–1511.
- [43] W. Wang, M.A. Carreira-Perpinán, Projection onto the probability simplex: An efficient algorithm with a simple proof, and an application, 2013, arXiv preprint arXiv:1309.1541.
- [44] X. Zhang, X. Xue, H. Sun, Z. Liu, L. Guo, X. Guo, Robust multiple kernel subspace clustering with block diagonal representation and low-rank consensus kernel, *Knowl.-Based Syst.* 227 (2021) 107243.
- [45] Z. Deng, K.-S. Choi, Y. Jiang, J. Wang, S. Wang, A survey on soft subspace clustering, *Inform. Sci.* 348 (2016) 84–106.
- [46] J.K. Benedetti, On the nonparametric estimation of regression functions, *J. R. Stat. Soc. Ser. B Stat. Methodol.* 39 (2) (1977) 248–253.
- [47] G. Karypis, CLUTO-A Clustering Toolkit, Tech. Rep., University of Minnesota, 2002.
- [48] E.Z. Macosko, A. Basu, R. Satija, J. Nemesk, K. Shekhar, M. Goldman, I. Tirosh, A.R. Bialas, N. Kamitaki, E.M. Martersteck, et al., Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets, *Cell* 161 (5) (2015) 1202–1214.
- [49] L. van der Maaten, G. Hinton, Visualizing data using t-SNE, *J. Mach. Learn. Res.* 9 (86) (2008) 2579–2605.