DISTRIBUTIONAL EQUIVALENCE OF KERNEL k-MEANS AND SPECTRAL CLUSTERING

Anonymous authors
Paper under double-blind review

ABSTRACT

The relationship between spectral clustering and kernel k-means has long been a subject of interest, but existing work primarily focuses on methodological connections, i.e., how to achieve the objectives of spectral clustering through kernel k-means or vice versa. This work, however, overlooks the fundamental connection between their objectives. This paper explores the relationship between their optimization objectives from a distributional perspective and provides a unified framework encompassing both. From a distributional perspective, kernel k-means aims to maximize the similarity of the distribution of each cluster, while spectral clustering also minimizes the similarity between the distribution of each cluster and the overall data distribution. Ncut combines these two through division, while Rcut uses subtraction. Based on this connection, a unified framework is proposed that simultaneously overcomes the shortcomings of both NCut and kernel k-means.

1 INTRODUCTION

Clustering, as an unsupervised machine learning algorithm Rokach & Maimon (2005), plays a crucial role in many fields, such as single-cell gene expression analysis Hu et al. (2021); Zhang et al. (2025), physicsMjahed (2006); Mikuni & Canelli (2021), and chemistryDowns & Barnard (2002); Consuegra-Jiménez et al. (2025). Since clustering methods lack labels, some methods utilize data density for clustering, such as the well-known DBSCAN Ester et al. (1996); Schubert et al. (2017) and Density Peaks Clustering (DPC) Rodriguez & Laio (2014), which often require density estimation. Some clustering methods are based on the similarity between data points, such as hierarchical clustering Murtagh & Contreras (2012; 2017), which does not require density estimation. In recent years, due to the powerful capabilities of deep neural networks, many neural network-based clustering methods have been proposed Xie et al. (2016); Cai et al. (2022); Li et al. (2021). These methods achieve better clustering performance by designing different optimization objectives, which are referred to as deep clustering Ren et al. (2024); Wei et al. (2024); Zhou et al. (2024). Optimization-based clustering algorithms using deep neural networks often require significant resources for training, while some optimization-based methods, such as k-means, kernel k-means, and spectral clustering, do not require extensive training.

Among so many clustering algorithms, kernel k-means Zhang & Rudnicky (2002); Filippone et al. (2008) and spectral clustering Von Luxburg (2007); Ng et al. (2001) have received widespread attention due to their sound theoretical foundation and clustering performance, especially for tabular datasets. The direct connection between kernel k-means and spectral clustering was first noticed by researchers two decades ago and has received continuous attention since Dhillon et al. (2004); Jiang et al. (2024); Jourdan et al. (2025). However, these studies are all confined to a single framework: how to achieve the optimization objective of spectral clustering through kernel k-means, or how to achieve the optimization objective of k-means through spectral clustering. In other words, they are all stuck on the methodological connections, neglecting the connection between their higher-level optimization objectives, which is shown in Figure 1.

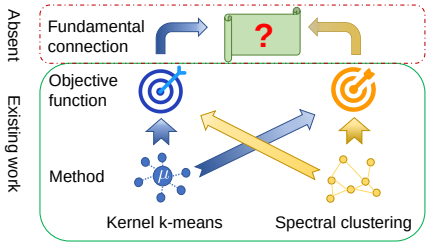

Figure 1: The connection between kernel k-means and spectral clustering.

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 paper aims to establish a connection between the two optimization objectives, rather than merely demonstrating their algorithmic equivalence. While a methodological link answers how to leverage one solver for the other's objective, it obscures the underlying principles that dictate why such methods are effective.

To establish a connection between the two clustering algorithms, we first discuss why it is currently difficult to establish such a connection and why researchers tend to overlook this more fundamental connection between the optimization objectives, focusing only on methodological connections. Our view is that this is because existing works currently approach these two optimization objectives from a point-based perspective. Therefore, this paper first rewrites both objectives from a distribution-based perspective. Then, we establish the connection between them from this distribution-based perspective, thereby establishing this important but previously long-absent fundamental connection between the optimization objectives.

The main contributions of this paper are threefold:

- Reformulating kernel k-means and spectral clustering from a distributional perspective
- Establishing the connection between kernel k-means and spectral clustering, demonstrating how the objective of one can be realized through the distributional perspective of the other.
- Developing a unified framework that simultaneously accommodates the objectives of kernel k-means and spectral clustering.

2 Related Work

2.1 Kernel k-means

Traditional k-means, while simple and efficient, are inherently based on Euclidean distance and can only handle linearly separable data structures Ahmed et al. (2020). To address this limitation, Kernel k-means was proposed to handle non-linear clustering problems. This method utilizes the kernel trick to project data points from the original input space into a high-dimensional or even infinite-dimensional reproducing kernel Hilbert space (RKHS) through a non-linear mapping $\phi(\cdot)$ Schölkopf et al. (1998). In this feature space, originally linearly inseparable data may become linearly separable, allowing for partitioning using the k-means approach. Girolami Girolami (2002) systematically described clustering methods based on Mercer kernels, proving that minimizing clustering error in the feature space is equivalent to maximizing the trace of the kernel matrix. However, directly calculating centroids in high-dimensional space is usually infeasible; therefore, kernel k-means is typically performed implicitly through calculations involving only dot products of the kernel matrix Tzortzis & Likas (2009). Although kernel k-means greatly expands the applicability of clustering, its computational complexity is mainly limited by the construction and storage of the kernel matrix, and it is prone to getting stuck in local optima.

2.2 Spectral Clustering

Spectral clustering, as a graph-based clustering method, has received widespread attention in recent years. Unlike centroid-based algorithms, spectral clustering constructs a similarity graph between data points and uses the spectrum (eigenvalues and eigenvectors) of the graph Laplacian matrix for dimensionality reduction and clustering Chung (1997). Shi and Malik Shi & Malik (2000) proposed the Normalized Cuts criterion and proved that the relaxed solution to this discrete optimization problem can be obtained by solving a generalized eigenvalue problem. Subsequently, Ng et al. (2001) proposed the classic NJW algorithm, which uses the first k eigenvectors of the Laplacian matrix to construct a new feature representation and applies k-means in this low-dimensional space. Because spectral clustering can capture the global structure on the data manifold, it performs excellently in handling data distributions with non-convex shapes (such as rings and spirals). Von Luxburg (2007) provided an in-depth theoretical analysis of spectral clustering from the perspectives of graph cuts and random walks, further solidifying its theoretical foundation.

2.3 Connection between Kernel k-means and Spectral Clustering

Although kernel k-means and spectral clustering originate from different theoretical backgrounds—the former from kernel methods and statistical learning, and the latter from graph theory and spectral graph theory—subsequent research has revealed a profound mathematical equivalence between them. The pioneering work Dhillon et al. (2004) formally established the theoretical link between these two methods. They proved that spectral clustering (specifically, normalized cut) can be reconstructed as a special weighted kernel k-means algorithm. Specifically, when the kernel matrix is derived from the adjacency matrix of the graph, and the weight matrix is set to the inverse of the graph's degree matrix, the objective function of weighted kernel k-means is equivalent to the relaxed objective function of spectral clustering. This finding not only explains why spectral clustering works effectively but also provides a theoretical framework for unifying these two methods. For example, subsequent methods using coresets to accelerate kernel k-means and spectral clustering are based on this connection. Furthermore, the connection between the two methods has been further analyzed from the perspective of coresets Jiang et al. (2024); Jourdan et al. (2025).

The classical result of Dhillon et al. (2004) supplies the operational equivalence and solver interchange. Degree-powered matrices are also related to existing normalization families: Coifman–Lafon reweight a kernel and then recompute degrees and Markov-normalize it for diffusion geometry Coifman & Lafon (2006), while regularized spectral clustering replaces D by $D + \tau I$ Qin & Rohe (2013). We do not claim these matrix operations as new. Our contribution is the distributional derivation that identifies degree as point-to-global-distribution similarity and yields the discrete degree-moment clustering objective between the two endpoints.

3 Re-formulation of Objectives

In this section, we will introduce distribution-based similarity and the definition of clustering from a distributional perspective. We will then provide a reinterpretation of both the kernel k-means algorithm and the spectral clustering algorithm from a distributional perspective. Finally, we will establish a connection between the two algorithms from this distributional perspective.

3.1 Cluster Similarity Based on Distribution Similarity.

Clustering algorithms often require measuring the point-cluster and cluster-cluster similarity. Both kernel k-means and spectral clustering algorithms are based on pairwise similarities between points, where the similarity between two points is defined as: $s(x, y) = \kappa(x, y)$, where κ is a kernel function. When calculating the similarity between a point and a cluster, or between two clusters, this is done by averaging the similarities between pairs of points:

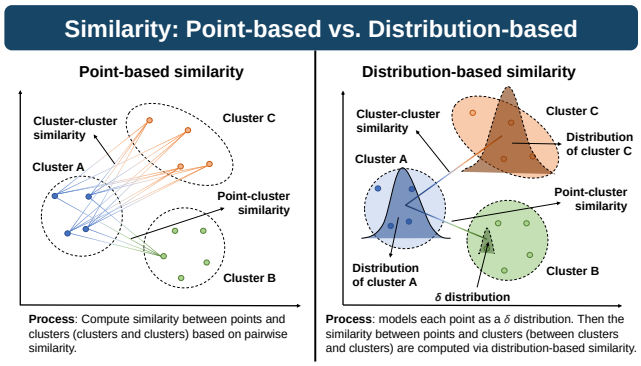

Figure 2: The comparison of point-based and distribution-based similarity.

$$s(x, C_i) = \frac{1}{|C_i|} \sum_{y \in C_i} \kappa(x, y)$$

and

$$s(C_i, C_j) = \frac{1}{|C_i| \times |C_j|} \sum_{x \in C_i, y \in C_j} \kappa(x, y), S(C_i, C_j) = |C_i| \times |C_j| \times s(C_i, C_j)$$

as shown in Figure 2.

This leads to the two following issues:

Issue 1: Computational complexity, because it requires calculating pairwise similarities, resulting in a complexity of $O(|C_i| \times |C_j|)$.

Issue 2: The similarity calculation is invariant to cluster membership: point pairs from the same cluster C_i and those from distinct clusters C_i and C_j are processed identically, implying that intra-cluster points are treated as mutually independent.

This stems from the underlying definition of clustering, which is based on points, meaning that points within the same cluster have higher pairwise similarity. To overcome the drawbacks of point-based similarity, we introduce a distribution-based metric, which includes the following definition:

Definition 3.1. The point x in cluster C_i is drawn from a distribution $\mathcal{P}_i$.

Definition 3.1 is an exact distributional reformulation: it does not alter pairwise similarities, but exposes within- and between-cluster relations through similarities between empirical distributions, supplying the distribution-level derivation missing from the pointwise view.

We will now demonstrate how Definition 3.1 addresses Issue 1 when the kernel has a finite-dimensional feature map[1]. Given dataset $D = \{C_1, C_2, \cdots, C_k\}$. The similarity between a point $x \in D$ and cluster C_i is: $s(x, C_i) = \kappa\left(\delta(x), \mathcal{P}_i\right)$, where δ is the Dirac delta function (or δ distribution), and the similarity between cluster C_i and cluster C_j is: $s(C_i, C_j) = \kappa\left(\mathcal{P}_i, \mathcal{P}_j\right)$.

In practice, we can employ kernel mean embedding (KME) to calculate similarity:

$$s(C_i, C_j) = \kappa(\mathcal{P}_i, \mathcal{P}_j) = \langle \mu_i, \mu_j \rangle \tag{1}$$

where $\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} \phi(x)$ and $\phi(\cdot)$ is the feature map of kernel function κ.

Based on Definition 3.1, if the kernel has an explicit or approximate m-dimensional feature map, the two mean embeddings and their inner product require $O((|C_i| + |C_j|)m)$ time. Exact evaluation for an implicit kernel such as the Gaussian kernel remains pairwise, and constructing its dense Gram matrix requires $O(|D|^2)$ storage and work; no linear-time claim is made for that setting.

3.2 Distribution-based Kernel k-means

The objective function of kernel k-means is based on point-based similarity, which is shown as follows: $J_k(D) = \sum_{i=1}^{k} \sum_{x \in C_i} \|\phi(x) - c_i\|^2$, where $c_i = \frac{\sum_{y \in C_i} \phi(y)}{|C_i|}$. We rewrite it as a distribution-based similarity function: $J_k(D) = \mathrm{Tr}(K) - \sum_{i=1}^{k} |C_i| \cdot \kappa(\mathcal{P}_i, \mathcal{P}_i)$. So the objective based on distribution is:

$$\min J_k(D) \Longleftrightarrow \max \sum_{i=1}^{k} |C_i| \cdot \kappa(\mathcal{P}_i, \mathcal{P}_i) \tag{2}$$

Here $\mathrm{Tr}(K)$ is independent of the partition and equals $|D|$ only when $K_{xx} = 1$ for every point, as in the Gaussian kernels used in our experiments.

From this reformulated distribution-based objective 2, we can see that kernel k-means aims to maximize the similarity within each cluster distribution itself, and uses the term $|C_i|$ to avoid generating clusters composed of only a few points.

3.3 Distribution-based Spectral Clustering

The objective function of spectral clustering (Ncut Shi (2003)) is based on point-based similarity, which is shown as follows: $J_n(D) = \sum_{i=1}^{k} \frac{S(C_i, D \setminus C_i)}{S(C_i, D)}$. We rewrite it as a distribution-based similarity function:

$$J_n(D) = \sum_{i=1}^{k} \left[1 - \frac{\kappa(\mathcal{P}_i, \mathcal{P}_i) \cdot |C_i|}{\kappa(\mathcal{P}_i, \mathcal{P}_D) \cdot |D|}\right], \ \min J_n(D) \Longleftrightarrow \max \sum_{i=1}^{k} \frac{\kappa(\mathcal{P}_i, \mathcal{P}_i) \cdot |C_i|}{\kappa(\mathcal{P}_i, \mathcal{P}_D) \cdot |D|} \tag{3}$$

From the reformulated distribution-based Ncut objective 3, we can see that Ncut, like kernel k-means, aims to maximize the similarity within each distribution, and uses the term $|C_i|$ to avoid generating

[1]Such as isolation kernel Ting et al. (2018) and the kernel functional approximation via Nyström method Williams & Seeger (2000) and Random Fourier Features Rahimi & Recht (2007).

clusters composed of only a few points. The difference is that Ncut also normalizes the similarity between the distribution of each cluster and the distribution of all the data, which increases the separation between clusters.

Table 1: Summary of objective based on distribution perspective (kkm refers to kernel k-means).

	Objective	Description	Reference
kkm	$\max \sum_{i=1}^{k} \lvert C_i \rvert \, \kappa(\mathcal{P}_i, \mathcal{P}_i)$	Maximize compactness.	Eq. 2
Ncut	$\max \sum_{i=1}^{k} \frac{\lvert C_i \rvert \, \kappa(\mathcal{P}_i, \mathcal{P}_i)}{\lvert D \rvert \, \kappa(\mathcal{P}_i, \mathcal{P}_D)}$	Maximize compactness and separation.	Eq. 3
Rcut	$\max \sum_{i=1}^{k} \left[\lvert C_i \rvert \, \kappa(\mathcal{P}_i, \mathcal{P}_i) - \lvert D \rvert \, \kappa(\mathcal{P}_i, \mathcal{P}_D) \right]$	Maximize compactness and separation.	Eq. 10

We summarize the objectives of kernel k-means, Ncut, and Rcut[2] Hagen & Kahng (1992) from the distribution perspective in Table 1. We have the following observations:

- Kernel k-means only maximizes the similarity within each cluster distribution and uses $|C_i|$ to prevent the formation of very small clusters.
- Both Ncut and Rcut consider not only the objective of kernel k-means (making the clusters more compact), but also minimize the similarity between each cluster distribution and the overall data distribution, which aims to increase the separation between clusters.
- The difference between Ncut and Rcut is that Ncut uses division for normalization, while subtraction is employed in Rcut.

4 Mutual Transformation and Implementation

In the previous section, we reformulated both kernel k-means and spectral clustering into distribution-based forms. In this section, we will explore the connection between the two based on this reformulation. Specifically, we will examine how to implement the objective function of one method using the other. This has been a key topic in existing work, as mentioned in the introduction. However, our approach here is different; we present this connection conceptually from a distributional perspective, rather than solely through mathematical notation, making it more intuitive.

4.1 Spectral Clustering using Weighted Kernel k-means

From the previous analysis, we already know that the difference between Ncut and kernel k-means lies in the normalization using the similarity between each distribution and the overall data distribution. To implement the Ncut optimization objective using kernel k-means, we set

$$\tilde{\phi}(x) \triangleq \frac{\phi(x)}{|D| \cdot \kappa(\delta(x), \mathcal{P}_D)}. \tag{4}$$

To obtain equal objectives, we need a new weight for each point $w(x) = |D| \cdot \kappa(\delta(x), \mathcal{P}_D)$, which in this case is essentially a weighted kernel k-means algorithm (see Appendix), and the weighted kernel k-means in this setting achieves the objective of Ncut. So: $J_W = \sum_{i=1}^{k} \frac{\sum_{x \in C_i} \sum_{y \in C_i} w(x) w(y) \langle \tilde{\phi}(x), \tilde{\phi}(y) \rangle}{\sum_{z \in C_i} w(z)} = \frac{Nu}{De}$. The numerator term is: $Nu = |C_i|^2 \cdot \kappa(\mathcal{P}_i, \mathcal{P}_i)$, and the denominator term is: $De = |C_i| \cdot |D| \cdot \kappa(\mathcal{P}_i, \mathcal{P}_D)$. Hence: $J_W = \sum_{i=1}^{k} \frac{|C_i| \cdot \kappa(\mathcal{P}_i, \mathcal{P}_i)}{|D| \cdot \kappa(\mathcal{P}_i, \mathcal{P}_D)}$.

[2]Detailed proofs can be found in the appendix. Unless otherwise specified, spectral clustering mentioned in this paper generally refers to Ncut.

In the previous sections, we established a direct connection between kernel k-means and Ncut based on their distribution representations. After achieving the objective of spectral clustering through weighted kernel k-means, we have a new connection: **spectral clustering (Ncut) is the kernel k-means with weighted** $w(x)$ that remaps (through mapping 4) the feature map of the samples in the reproducing kernel Hilbert space after being mapped by the kernel function, using the similarity between each point and the overall data distribution. Therefore, the term $|D| \cdot \kappa(\mathcal{P}_i, \mathcal{P}_D)$ in spectral clustering serves two purposes: re-mapping and weighting.

4.2 Kernel k-means Using Spectral Clustering

The reverse direction is exact under a stated constant-degree condition. Let $K \succeq 0$ be entrywise nonnegative, and define $\mathrm{RAssoc}_K(\mathcal{C}) = \sum_{i=1}^{k} \frac{\mathbf{1}_{C_i}^\top K \mathbf{1}_{C_i}}{|C_i|}$, $J_{\mathrm{KKM}}^K(\mathcal{C}) = \mathrm{Tr}(K) - \mathrm{RAssoc}_K(\mathcal{C})$.

Proposition 4.1 (Constant-degree reverse equivalence)**.** *Suppose $K\mathbf{1} = d_0\mathbf{1}$ for $d_0 > 0$, and let $\widetilde{K} = d_0 K$. Then $\widetilde{K} \succeq 0$, and for every k-partition $\mathcal{C}$,*

$$J_{\mathrm{Ncut}}^{\widetilde{K}}(\mathcal{C}) = k - \frac{\mathrm{Tr}(K)}{d_0} + \frac{J_{\mathrm{KKM}}^K(\mathcal{C})}{d_0}. \tag{5}$$

Consequently, the two discrete objectives have exactly the same minimizers.

Indeed, every degree of $\widetilde{K}$ equals d_0^2, so $\mathrm{NAssoc}_{\widetilde{K}}(\mathcal{C}) = d_0^{-1}\,\mathrm{RAssoc}_K(\mathcal{C})$; substituting $J_{\mathrm{Ncut}} = k - \mathrm{NAssoc}$ proves Equation 5. Thus constant degree is the mechanism by which volume normalization becomes partition independent. It is not an assumption of the unified framework or of the experiments. On a regular graph, the family in Section 5 satisfies $Q_\lambda = d_0^{-\lambda} Q_0$; on the heterogeneous-degree graphs used in the experiments, the degree moment varies with λ. Appendix B.2 gives a partition-independent construction for general nonnegative PSD K at the level of discrete objective ordering.

Table 2: Mathematical equivalence between Ncut and kernel k-means and the scope of each direction.

Target Objective	**Feature** $\tilde{\phi}(x)$	**Remapping**	**Implementation Conditions**
Ncut using kernel k-means	$\tilde{\phi}(x) \triangleq \frac{\phi(x)}{\lvert D\rvert \cdot \kappa(\delta(x), \mathcal{P}_D)}$		$w(x) = \lvert D\rvert \cdot \kappa(\delta(x), \mathcal{P}_D)$
kernel k-means using Ncut	$\widetilde{K} = d_0 K$		$K\mathbf{1} = d_0\mathbf{1}$; positive-affine equality in Equation 5

The constant-degree condition is used only for the reverse construction; it is not required by the unified objective or the experiments.

5 A Unified Framework for Kernel k-means and Spectral Clustering.

It is observed that both kernel k-means, Ncut, and Rcut all include two important terms: $\kappa(\mathcal{P}_i, \mathcal{P}_i)|C_i|$ and $\kappa(\mathcal{P}_i, \mathcal{P}_D)|D|$. Therefore, they can be uniformly represented as

$$\mathcal{Q}_{f,g}(D) = \sum_{i=1}^{k} f\left(\kappa(\mathcal{P}_i, \mathcal{P}_i)|C_i|, g_\lambda(\mathcal{P}_i, \mathcal{P}_D)\right). \tag{6}$$

The essence of $\mathcal{Q}(D)$ is the assumption that each cluster is generated by a distribution P_i, and the goal is to assign each point to the cluster corresponding to the distribution from which it originated. This allows for higher intra-cluster compactness and greater inter-cluster separation.

By setting different functions $f(\cdot,\cdot)$ and $g_\lambda(\cdot,\cdot)$, different objectives can be obtained. For the continuum between kernel k-means and Ncut, we take $f(a,b) = a/b$. We first transform the point-to-distribution similarity by the power λ and then aggregate it over the cluster distribution, as defined below.

Assuming positive degrees, define

$$S(x,D)=\sum_{y\in D}\kappa(x,y)=|D|\kappa(\delta(x),\mathcal{P}_D),\quad S_\lambda(C_i,D)=\sum_{x\in C_i}S(x,D)^\lambda=\mathbf{1}_{C_i}^\top\mathbf{D}^\lambda\mathbf{1}_{C_i},$$
$$g_\lambda(\mathcal{P}_i,\mathcal{P}_D)=\frac{S_\lambda(C_i,D)}{|C_i|},\qquad \mathbf{D}=\mathrm{diag}(S(x_1,D),\dots,S(x_n,D)). \tag{7}$$

At the endpoints, $g_0(\mathcal{P}_i,\mathcal{P}_D)=1$ and $g_1(\mathcal{P}_i,\mathcal{P}_D)=|D|\kappa(\mathcal{P}_i,\mathcal{P}_D)$. For $0<\lambda<1$, the order of transformation and distributional aggregation is consequential. The alternative order, in which the cluster mean is formed before applying the power, defines a distinct interior objective; Appendix C.2 gives its formulation and a corresponding discrete optimizer.

With $f(a,b)=a/b$, Equation 6 becomes

$$Q_\lambda(D)=\sum_{i=1}^k\frac{|C_i|\kappa(\mathcal{P}_i,\mathcal{P}_i)}{g_\lambda(\mathcal{P}_i,\mathcal{P}_D)}=\sum_{i=1}^k\frac{S(C_i,C_i)}{S_\lambda(C_i,D)}=\sum_{i=1}^k\frac{\mathbf{1}_{C_i}^\top K\mathbf{1}_{C_i}}{\mathbf{1}_{C_i}^\top\mathbf{D}^\lambda\mathbf{1}_{C_i}}. \tag{8}$$

Thus Q_0 is the kernel k-means association objective and Q_1 is normalized association, equivalently Ncut up to the additive constant k.

For $h_i=\mathbf{1}_{C_i}/\sqrt{S_\lambda(C_i,D)}$ and $H=[h_1,\dots,h_k]$, we have $H^\top\mathbf{D}^\lambda H=I$ and $Q_\lambda(D)=\mathrm{Tr}(H^\top KH)$. Relaxing the partition-indicator constraints while retaining $H^\top\mathbf{D}^\lambda H=I$, and substituting $\widetilde{H}=\mathbf{D}^{\lambda/2}H$, gives

$$\max_{\widetilde{H}^\top\widetilde{H}=I}\mathrm{Tr}\Big(\widetilde{H}^\top\mathbf{D}^{-\lambda/2}K\mathbf{D}^{-\lambda/2}\widetilde{H}\Big). \tag{9}$$

By Rayleigh–Ritz Lütkepohl (1997), the relaxed solution uses the leading k eigenvectors of $\mathbf{D}^{-\lambda/2}K\mathbf{D}^{-\lambda/2}$, exactly as in Algorithm 1. Row normalization and k-means provide the rounding used throughout the experiments.

The algorithm for the unified framework, termed λ-spectral clustering[3], is shown in Algorithm 1.

Algorithm 1 λ-Spectral Clustering

Require: Data set $D=\{x_1,\dots,x_n\}$, number of clusters k, parameter $\lambda\in[0,1]$, kernel function $\kappa(\cdot,\cdot)$.
Ensure: Cluster partition $\mathcal{C}=\{C_1,\dots,C_k\}$.
1: Compute K, $d_i=\sum_{j=1}^n K_{ij}$ and $W_{ii}=(d_i)^\lambda$.
2: Compute the L_{sym} matrix $L_{sym}=W^{-1/2}KW^{-1/2}$
3: Find the k eigenvectors $U=[u_1,\dots,u_k]\in\mathbb{R}^{n\times k}$ corresponding to the largest eigenvalues of L_{sym}.
4: Normalize the rows of U: $Y_{ij}=\frac{U_{ij}}{\sqrt{\sum_{l=1}^k U_{il}^2}}$.
5: Treat each row of Y as a point in $\mathbb{R}^k$ and apply k-means to cluster them into partition $\mathcal{C}$.

6 Experiments

In this section, we will demonstrate the effect of λ on λ-spectral clustering. To analyze the impact of λ on λ-spectral clustering, we used two synthetic datasets and six benchmark datasets[4].

The original experiments were executed on a Linux machine with 1.175 TB RAM and a 128-core AMD CPU at 2 GHz.

6.1 Synthetic Datasets

The synthetic datasets and the clustering results using λ-spectral clustering under different λ values ($\lambda\in\{0,0.5,1.0\}$) are shown in Figure 3. The OUTLIER dataset consists of two clusters and also

[3]The source code is available in `https://anonymous.4open.science/r/LambdaSC`.
[4]The characteristics are shown in the Appendix.

contains several outliers in the upper right corner. The LOW-HIGH dataset consists of two clusters, one small and one large, with the smaller cluster having a higher density of points relative to the edges of the larger cluster.

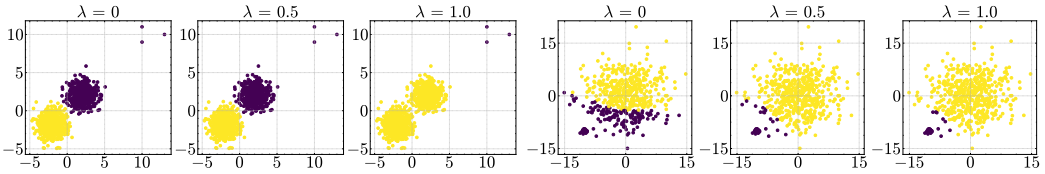

Figure 3: The clustering results of two Synthetic datasets (OUTLIER (left) and LOW-HIGH (right)).

For the OUTLIER dataset, which contains outliers, the λ-spectral clustering performs well when $\lambda = 0$ (corresponding to kernel k-means), correctly clustering the data. However, when $\lambda = 1$ (corresponding to Ncut), it clusters the outliers into one cluster and the remaining data into another. This is a well-known drawback of spectral clustering, namely its sensitivity to outliers. This is easily understood from a distributional perspective, as these outliers have low similarity to the overall distribution, since outliers are often considered to originate from a different distribution. This leads to the outliers being grouped into a single cluster after $\kappa(\mathcal{P}_i, \mathcal{P}_D) \cdot |D|$ normalization, which maximizes $\mathcal{Q}(D)$.

The λ-spectral clustering performs poorly on the LOW-HIGH dataset when $\lambda = 0$. This is due to the absence of $\kappa(\mathcal{P}_i, \mathcal{P}_D) \cdot |D|$, which causes points from the larger cluster near the relatively high-density cluster in the lower left corner to be incorrectly assigned to that cluster. As λ increases, the clustering performance improves, and more and more points are correctly clustered into the upper right cluster.

By employing our proposed λ-Spectral clustering method (for instance, with λ set to 0.5), we simultaneously overcome the limitations of both spectral clustering and k-means; consequently, on both datasets, we avoid the poor clustering results typically caused by outliers and data imbalance.

6.2 Benchmark Datasets

The results of λ-spectral clustering in terms of Normalized Mutual Information (NMI) McDaid et al. (2011) and Adjusted Rand Index (ARI) Rand (1971) on six benchmark datasets with various λ values ($\lambda \in \{0, 0.05, \ldots, 1.0\}$) are shown in Figure 4.

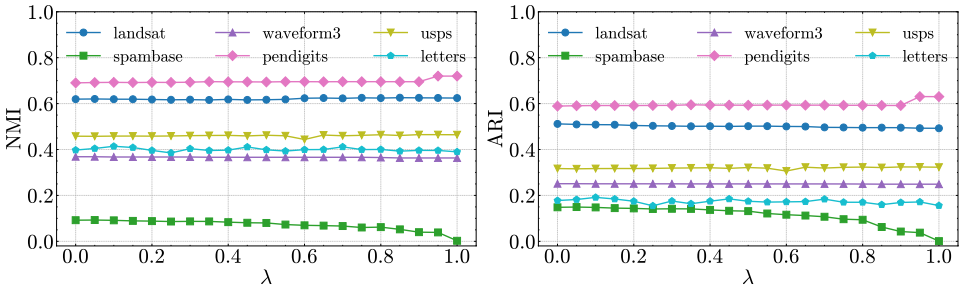

Figure 4: Clustering results on benchmark dataset.

On the four datasets other than spambase and pendigits, λ-spectral clustering showed slight fluctuations but no significant changes as λ varied. However, the opposite trend was observed on the spambase and pendigits datasets: on spambase, the clustering performance worsened as λ increased, while on the pendigits dataset, the clustering performance improved as λ ($\lambda > 0.9$) increased.

To analyze the reasons for these different trends, we plotted the density (similarity $\kappa(\delta(x), \mathcal{P}_D)$) of points in each cluster of the dataset, as shown in Figure 7. Figure 7 shows that both clusters in the spambase dataset contain points with very low density (which can be considered outliers), while the pendigits dataset contains clusters with varied density (e.g., clusters 5 and 8 contain a large number of low-density points, while the points in cluster 7 have a higher density).

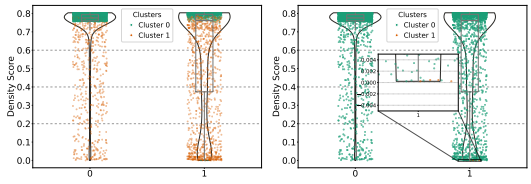

Figure 5: Density of spambase (left: λ=0, right: λ=1.0).

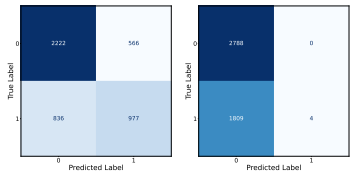

Figure 6: Confusion matrix of spambase (left: λ=0, right: λ=1.0).

For the Spambase dataset (as shown in Figure 5), due to the presence of outliers, when λ is very small, λ-spectral clustering incorrectly clusters these low-density points into a single cluster. As λ increases, when it reaches 1.0, it even more drastically clusters a few low-density points (4 points) into one cluster, similar to what happens with the OUTLIER dataset. This can also be observed in the confusion matrix shown in Figure 6.

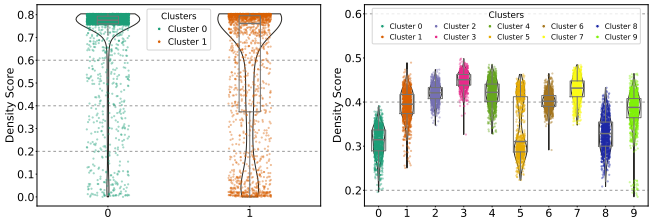

Figure 7: Density of spambase (left) and pendigits (right).

For the pendigits dataset, when $\lambda = 0$, due to the presence of clusters with varying densities in the data, as shown in Figure 8, some points in less dense clusters are incorrectly assigned to other clusters. For example, the low-density points in cluster 5 are separated into a single cluster, and the points in cluster 8 are incorrectly assigned to the very dense cluster 7. As λ increases to 1.0, although the points in cluster 5 are still misclassified, most of the points in cluster 8 are correctly clustered.

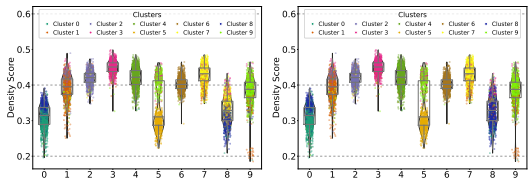

Figure 8: Density of pendigits (left: λ=0, right: λ=1.0).

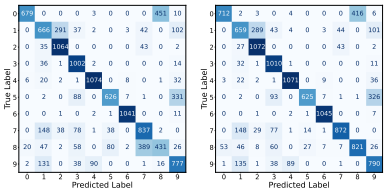

Figure 9: Confusion matrix of pendigits (left: λ=0, right: λ=1.0).

This is consistent with the LOW-HIGH dataset, and the same result can be observed in Figure 9. When $\lambda = 0$, 389 points from cluster 8 are misclassified into cluster 7, whereas at $\lambda = 1.0$, the number of misclassified points is significantly reduced, only 7 points are misclassified into cluster 7.

7 Conclusion

This paper unveils a fundamental connection between kernel k-means and spectral clustering by reformulating their objectives from a point-wise similarity view to a distributional perspective. Specifically, we demonstrate that while kernel k-means maximizes the distributional similarity within each cluster, spectral clustering normalizes this term against the similarity between cluster distributions and the global data distribution. Therefore, by remapping the feature map of each point based on the similarity between individual point distributions and the global distribution, we prove that the objectives of these two methods are mutually recoverable under specific constraints. This insight leads to a unified framework where a tunable parameter, λ, facilitates the interpolation between the two objectives, allowing us to analyze the algorithmic impact of varying λ. We demonstrated the impact of different λ values on clustering results through experiments on synthetic and benchmark datasets.

AI USE STATEMENT

Generative AI assisted with language editing, literature and reference checks, and the checking of theoretical statements and proofs. This introduced no new experimental measurements. The authors retain responsibility for the final manuscript, including verification of all AI-assisted content.

ETHICS STATEMENT

This work concerns unsupervised clustering of benchmark feature matrices. No new human-subject data were collected. Clustering can amplify representation bias, so deployment in consequential settings requires dataset-specific fairness, privacy, and error analysis.

REFERENCES

Mohiuddin Ahmed, Raihan Seraj, and Syed Mohammed Shamsul Islam. The k-means algorithm: A comprehensive survey and performance evaluation. *Electronics*, 9(8):1295, 2020.

E. Alpaydin and Fevzi. Alimoglu. Pen-Based Recognition of Handwritten Digits. UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5MG6K.

Deng Cai, Xiaofei He, Jiawei Han, and Thomas S Huang. Graph regularized nonnegative matrix factorization for data representation. *IEEE transactions on pattern analysis and machine intelligence*, 33(8):1548–1560, 2010.

Jinyu Cai, Jicong Fan, Wenzhong Guo, Shiping Wang, Yunhe Zhang, and Zhao Zhang. Efficient deep embedded subspace clustering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1–10, 2022.

Fan RK Chung. *Spectral graph theory*, volume 92. American Mathematical Soc., 1997.

Ronald R. Coifman and Stéphane Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30, 2006.

Sebastián Consuegra-Jiménez, Cesar Tovio-Gracia, and Ricardo Vivas-Reyes. Unsupervised learning techniques for clustering analysis of physicochemical properties in the periodic table elements. *Results in Chemistry*, pp. 102517, 2025.

Inderjit S Dhillon, Yuqiang Guan, and Brian Kulis. Kernel k-means: spectral clustering and normalized cuts. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 551–556, 2004.

Geoff M Downs and John M Barnard. Clustering methods and their uses in computational chemistry. *Reviews in computational chemistry*, 18:1–40, 2002.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pp. 226–231, 1996.

Maurizio Filippone, Francesco Camastra, Francesco Masulli, and Stefano Rovetta. A survey of kernel and spectral methods for clustering. *Pattern recognition*, 41(1):176–190, 2008.

Peter W Frey and David J Slate. Letter recognition using holland-style adaptive classifiers. *Machine learning*, 6:161–182, 1991.

Mark Girolami. Mercer kernel-based clustering in feature space. *IEEE transactions on neural networks*, 13(3):780–784, 2002.

Lars Hagen and Andrew B Kahng. New spectral methods for ratio cut partitioning and clustering. *IEEE transactions on computer-aided design of integrated circuits and systems*, 11(9):1074–1085, 1992.

Mark Hopkins, Erik Reeber, George Forman, and Jaap Suermondt. Spambase. UCI Machine Learning Repository, 1999. DOI: https://doi.org/10.24432/C53G6X.

Jian Hu, Xiangjie Li, Kyle Coleman, Amelia Schroeder, Nan Ma, David J Irwin, Edward B Lee, Russell T Shinohara, and Mingyao Li. Spagcn: Integrating gene expression, spatial location and histology to identify spatial domains and spatially variable genes by graph convolutional network. *Nature methods*, 18(11):1342–1351, 2021.

Shaofeng H-C Jiang, Robert Krauthgamer, Jianing Lou, and Yubo Zhang. Coresets for kernel clustering. *Machine Learning*, 113(8):5891–5906, 2024.

Ben Jourdan, Gregory Schwartzman, Peter Macgregor, and He Sun. Coreset spectral clustering. In *The Thirteenth International Conference on Learning Representations*, 2025.

Yunfan Li, Peng Hu, Zitao Liu, Dezhong Peng, Joey Tianyi Zhou, and Xi Peng. Contrastive clustering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 8547–8555, 2021.

Helmut Lütkepohl. *Handbook of matrices*. John Wiley & Sons, 1997.

Aaron F McDaid, Derek Greene, and Neil Hurley. Normalized mutual information to evaluate overlapping community finding algorithms. *arXiv preprint arXiv:1110.2515*, 2011.

Vinicius Mikuni and Florencia Canelli. Unsupervised clustering for collider physics. *Physical Review D*, 103(9):092007, 2021.

Mostafa Mjahed. The use of clustering techniques for the classification of high energy physics data. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 559(1):199–202, 2006.

Fionn Murtagh and Pedro Contreras. Algorithms for hierarchical clustering: an overview. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, 2(1):86–97, 2012.

Fionn Murtagh and Pedro Contreras. Algorithms for hierarchical clustering: an overview, ii. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 7(6):e1219, 2017.

Andrew Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 14, 2001.

Tai Qin and Karl Rohe. Regularized spectral clustering under the degree-corrected stochastic blockmodel. In *Advances in Neural Information Processing Systems*, volume 26, 2013.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. *Advances in neural information processing systems*, 20, 2007.

William M Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971.

Yazhou Ren, Jingyu Pu, Zhimeng Yang, Jie Xu, Guofeng Li, Xiaorong Pu, Philip S Yu, and Lifang He. Deep clustering: A comprehensive survey. *IEEE transactions on neural networks and learning systems*, 36(4):5858–5878, 2024.

Alex Rodriguez and Alessandro Laio. Clustering by fast search and find of density peaks. *science*, 344(6191):1492–1496, 2014.

Lior Rokach and Oded Maimon. Clustering methods. In *Data mining and knowledge discovery handbook*, pp. 321–352. Springer, 2005.

Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319, 1998.

Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu. Dbscan revisited, revisited: why and how you should (still) use dbscan. *ACM Transactions on Database Systems (TODS)*, 42(3):1–21, 2017.

Shi. Multiclass spectral clustering. In *Proceedings ninth IEEE international conference on computer vision*, pp. 313–319. IEEE, 2003.

Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

Kai Ming Ting, Yue Zhu, and Zhi-Hua Zhou. Isolation kernel and its effect on svm. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2329–2337, 2018.

Grigorios F Tzortzis and Aristidis C Likas. The global kernel k-means algorithm for clustering in feature space. *IEEE transactions on neural networks*, 20(7):1181–1194, 2009.

Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

Yizhang Wang, Jiaxin Qian, Muhammad Hassan, Xinyu Zhang, Tao Zhang, Chao Yang, Xingxing Zhou, and Fengjin Jia. Density peak clustering algorithms: A review on the decade 2014–2023. *Expert Systems with Applications*, 238:121860, 2023.

Xiuxi Wei, Zhihui Zhang, Huajuan Huang, and Yongquan Zhou. An overview on deep clustering. *Neurocomputing*, 590:127761, 2024.

Christopher Williams and Matthias Seeger. Using the nyström method to speed up kernel machines. *Advances in neural information processing systems*, 13, 2000.

Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pp. 478–487. PMLR, 2016.

Hang Zhang, Yi Zhang, Kai Ming Ting, Jie Zhang, and Qiuran Zhao. Kernel-bounded clustering for spatial transcriptomics enables scalable discovery of complex spatial domains. *Genome Research*, 35(2):355–367, 2025.

Rong Zhang and Alexander I Rudnicky. A large scale clustering scheme for kernel k-means. In *2002 International conference on pattern recognition*, volume 4, pp. 289–292. IEEE, 2002.

Sheng Zhou, Hongjia Xu, Zhuonan Zheng, Jiawei Chen, Zhao Li, Jiajun Bu, Jia Wu, Xin Wang, Wenwu Zhu, and Martin Ester. A comprehensive survey on deep clustering: Taxonomy, challenges, and future directions. *ACM Computing Surveys*, 57(3):1–38, 2024.

A Details of the Re-formulation

To establish a rigorous distributional connection between spectral clustering and kernel k-means, we must formally reformulate their optimization objectives. Rather than relying on traditional point-based similarity metrics, which often obscure the underlying statistical mechanics, we use a distribution-based formulation. This shift in perspective allows us to treat discrete clusters as continuous probability distributions, providing a deeper and more unified theoretical link.

A.1 Re-formulation of Kernel k-means

The traditional kernel k-means objective minimizes the sum of squared distances between individual data points and their assigned cluster centroids in the Reproducing Kernel Hilbert Space (RKHS). We can expand this Euclidean distance and express it entirely in terms of the kernel mean embedding μ_i. By doing so, the objective seamlessly transforms into a maximization of intra-cluster distributional similarity.

Distribution-based kernel k-means:

$$
\begin{aligned}
J_k(D) &= \sum_{i=1}^{k} \sum_{x \in C_i} \|\phi(x) - c_i\|^2 \\
&= \sum_{i=1}^{k} \sum_{x \in C_i} \left[\langle \phi(x), \phi(x) \rangle - \frac{2 \sum_{y \in C_i} \langle \phi(x), \phi(y) \rangle}{|C_i|} + \langle \frac{\sum_{y \in C_i} \phi(y)}{|C_i|}, \frac{\sum_{z \in C_i} \phi(z)}{|C_i|} \rangle \right] \\
&= \sum_{i=1}^{k} \sum_{x \in C_i} \left[\kappa(x, x) - \frac{2 \sum_{y \in C_i} \langle \phi(x), \phi(y) \rangle}{|C_i|} + \langle \mu_i, \mu_i \rangle \right] \\
&= \sum_{i=1}^{k} \sum_{x \in C_i} \left[\kappa(x, x) - 2 \langle \phi(x), \mu_i \rangle + \langle \mu_i, \mu_i \rangle \right] \\
&= \mathrm{Tr}(K) + \sum_{i=1}^{k} \left[-2 |C_i| \langle \frac{\sum_{x \in C_i} \phi(x)}{|C_i|}, \mu_i \rangle + |C_i| \langle \mu_i, \mu_i \rangle \right] = \mathrm{Tr}(K) - \sum_{i=1}^{k} |C_i| \langle \mu_i, \mu_i \rangle \\
&= \mathrm{Tr}(K) - \sum_{i=1}^{k} |C_i| \cdot \kappa(\mathcal{P}_i, \mathcal{P}_i)
\end{aligned}
$$

This final expression clearly demonstrates that minimizing the traditional k-means error is mathematically identical to maximizing the structural compactness of each cluster's probability distribution.

A.2 Re-formulation of Spectral Clustering

In the context of spectral clustering, the Normalized Cut objective tries to partition the graph such that the cut between clusters is minimized relative to the total volume of the clusters. From a distributional standpoint, we can interpret this as a mechanism that balances the intra-cluster similarity against the global data distribution.

Distribution-based spectral clustering:

$$
\begin{aligned}
J_n(D) &= \sum_{i=1}^{k} \frac{S(C_i, D \setminus C_i)}{S(C_i, D)} = \sum_{i=1}^{k} \frac{S(C_i, D) - S(C_i, C_i)}{S(C_i, D)} \\
&= \sum_{i=1}^{k} \frac{\langle \mu_i, \mu_D \rangle |C_i| \, |D| - \langle \mu_i, \mu_i \rangle |C_i| |C_i|}{\langle \mu_i, \mu_D \rangle |C_i| \, |D|} \\
&= \sum_{i=1}^{k} \left[\frac{\langle \mu_i, \mu_D \rangle |C_i| \, |D|}{\langle \mu_i, \mu_D \rangle |C_i| \, |D|} - \frac{\langle \mu_i, \mu_i \rangle |C_i| |C_i|}{\langle \mu_i, \mu_D \rangle |C_i| \, |D|} \right] = \sum_{i=1}^{k} \left[1 - \frac{\kappa(\mathcal{P}_i, \mathcal{P}_i) \cdot |C_i|}{\kappa(\mathcal{P}_i, \mathcal{P}_D) \cdot |D|} \right]
\end{aligned}
$$

Here, the normalization factor $\kappa(\mathcal{P}_i, \mathcal{P}_D) \cdot |D|$ acts as a structural penalty. It penalizes clusters that are too similar to the global distribution, thereby actively encouraging distinct and well-separated clusters.

A.3 Ratio Cut

The Ratio Cut objective differs from Ncut by penalizing the absolute size of the clusters rather than their volume. Notably, we explicitly reject the standard matrix-based derivations of the graph Laplacian that dominate existing literature. Instead, we establish the theoretical equivalence strictly through a specific distribution-based mapping.

We express the distribution-based Ratio Cut as follows.

$$\begin{aligned} J_r(D) &= \sum_{i=1}^{k} \frac{S(C_i, D \setminus C_i)}{|C_i|} = \sum_{i=1}^{k} \frac{S(C_i, D) - S(C_i, C_i)}{|C_i|} \\ &= \sum_{i=1}^{k} \frac{\langle \mu_i, \mu_D \rangle |C_i| \, |D| - \langle \mu_i, \mu_i \rangle |C_i|^2}{|C_i|} = \sum_{i=1}^{k} \left[\langle \mu_i, \mu_D \rangle \, |D| - \langle \mu_i, \mu_i \rangle |C_i| \right] \\ &= \sum_{i=1}^{k} \left[\kappa(\mathcal{P}_i, \mathcal{P}_D) \cdot |D| - \kappa(\mathcal{P}_i, \mathcal{P}_i) \cdot |C_i| \right] \end{aligned}$$

$$\min J_r(D) \Longleftrightarrow \max \sum_{i=1}^{k} \left[\kappa(\mathcal{P}_i, \mathcal{P}_i) \cdot |C_i| - \kappa(\mathcal{P}_i, \mathcal{P}_D) \cdot |D| \right] \tag{10}$$

B Details of Mutual Transformation and Implementation

Building upon the reformulated objectives, we now detail the specific mathematical mechanics required to implement one algorithm using the operational framework of the other. The core technique involves remapping the feature space and introducing targeted point-wise weights.

B.1 Spectral Clustering using Weighted Kernel k-means

To successfully achieve the Ncut objective via kernel k-means, we must mathematically compensate for the global distribution normalization inherent to Ncut. We achieve this by introducing a point-specific weight $w(x)$ that perfectly reflects the density of the global distribution at that specific point. The objective for the weighted kernel k-means is formulated over the remapped feature space $\tilde{\phi}(x)$.

Spectral clustering using weighted kernel k-means:

$$J_W = \sum_{i=1}^{k} \frac{\sum_{x \in C_i} \sum_{y \in C_i} w(x) w(y) \langle \tilde{\phi}(x), \tilde{\phi}(y) \rangle}{\sum_{z \in C_i} w(z)} = \frac{Nu}{De}$$

The numerator evaluates the pairwise similarities within the newly remapped space.

$$\begin{aligned} Nu &= \sum_{x \in C_i} \sum_{y \in C_i} \left[|D| \, \kappa(\delta(x), \mathcal{P}_D) \cdot |D| \, \kappa(\delta(y), \mathcal{P}_D) \cdot \frac{\langle \phi(x), \phi(y) \rangle}{|D|^2 \, \kappa(\delta(x), \mathcal{P}D) \kappa(\delta(y), \mathcal{P}D)} \right] \\ &= \sum_{x \in C_i} \sum_{y \in C_i} \kappa(\delta(x), \delta(y)) = |C_i|^2 \cdot \kappa(\mathcal{P}_i, \mathcal{P}_i) \end{aligned}$$

Meanwhile, the denominator seamlessly aggregates the density weights across the entire cluster.

$$De = \sum_{z \in C_i} w(z) = \sum_{z \in C_i} |D| \cdot \kappa(\delta(z), \mathcal{P}_D) = |C_i| \cdot |D| \cdot \kappa(\mathcal{P}_i, \mathcal{P}_D)$$

Combining these isolated terms yields the target equivalence:

$$J_W = \sum_{i=1}^{k} \frac{|C_i|^2 \cdot \kappa(\mathcal{P}_i, \mathcal{P}_i)}{|C_i| \cdot |D| \cdot \kappa(\mathcal{P}_i, \mathcal{P}_D)} = \sum_{i=1}^{k} \frac{|C_i| \cdot \kappa(\mathcal{P}_i, \mathcal{P}_i)}{|D| \cdot \kappa(\mathcal{P}_i, \mathcal{P}_D)}$$

This confirms that appropriately weighting the individual points and remapping the kernel function allows standard k-means to inherently execute spectral clustering.

B.2 Kernel k-means Using Spectral Clustering

Under Proposition 4.1, $\widetilde{K} = d_0 K$ has degree d_0^2 at every point. Therefore,

$$\mathrm{NAssoc}_{\widetilde{K}}(\mathcal{C}) = \sum_{i=1}^{k} \frac{d_0\, \mathbf{1}_{C_i}^\top K \mathbf{1}_{C_i}}{d_0^2 |C_i|} = \frac{1}{d_0}\, \mathrm{RAssoc}_K(\mathcal{C}).$$

Using $J^K_{\mathrm{KKM}} = \mathrm{Tr}(K) - \mathrm{RAssoc}_K$ and $J_{\mathrm{Ncut}} = k - \mathrm{NAssoc}$ proves Equation 5. The realized Gram matrix is PSD because it is a positive scalar multiple of K.

A partition-independent discrete construction. Let K be a nonzero, entrywise nonnegative, symmetric PSD matrix, $d = K\mathbf{1}$, and $\bar{d} = |D|^{-1}\mathbf{1}^\top d$. Choose

$$c > 0, \qquad c \geq \max_x (2d_x - \bar{d}),$$

and define

$$q = \frac{1}{|D|}\left(\frac{c + \bar{d}}{2}\mathbf{1} - d\right), \qquad A = K + q\mathbf{1}^\top + \mathbf{1}q^\top.$$

Then A is symmetric and entrywise nonnegative, $A\mathbf{1} = c\mathbf{1}$, and, for every partition $\mathcal{C}$,

$$\mathrm{NAssoc}_A(\mathcal{C}) = \frac{1}{c}\, \mathrm{RAssoc}_K(\mathcal{C}) + 1 - \frac{\bar{d}}{c}. \tag{11}$$

Indeed, the bound on c gives $q_x \geq 0$ and $\mathbf{1}^\top q = (c - \bar{d})/2$. For each cluster C, $\mathbf{1}_C^\top A \mathbf{1}_C = \mathbf{1}_C^\top K \mathbf{1}_C + 2|C|\mathbf{1}_C^\top q$; division by $c|C|$ and summation prove Equation 11. Thus Ncut on A and kernel k-means on K have the same discrete optimizers. This optional construction claims only positive-affine equivalence of the discrete objectives: A is not asserted to be PSD, and no equality of relaxations, eigenspaces, or rounded partitions is claimed.

B.3 Weighted Kernel k-means

The objective of weighted kernel k-means is:

$$J(D) = W - \sum_{i=1}^{k} W_i \cdot \kappa_w(\mathcal{P}_i, \mathcal{P}_i).$$

$$
\begin{aligned}
J_W(D) &= \sum_{i=1}^{k} \sum_{x \in C_i} w(x) |\phi(x) - c_i|^2 = \sum_{i=1}^{k} \sum_{x \in C_i} w(x) \left[\langle \phi(x), \phi(x) \rangle \right. \\
&\quad \left. - \frac{2 \sum_{y \in C_i} w(y) \langle \phi(x), \phi(y) \rangle}{W_i} + \langle \frac{\sum_{y \in C_i} w(y) \phi(y)}{W_i}, \frac{\sum_{z \in C_i} w(z) \phi(z)}{W_i} \rangle \right] \\
&= \sum_{i=1}^{k} \sum_{x \in C_i} w(x) \left[1 - \frac{2 \sum_{y \in C_i} w(y) \langle \phi(x), \phi(y) \rangle}{W_i} + \langle \mu_i, \mu_i \rangle \right] \\
&= \sum_{i=1}^{k} \sum_{x \in C_i} w(x) \left[1 - 2 \langle \phi(x), \mu_i \rangle + \langle \mu_i, \mu_i \rangle \right] \\
&= W + \sum_{i=1}^{k} \left[-2 W_i \langle \frac{\sum_{x \in C_i} w(x) \phi(x)}{W_i}, \mu_i \rangle + W_i \langle \mu_i, \mu_i \rangle \right] \\
&= W - \sum_{i=1}^{k} W_i \langle \mu_i, \mu_i \rangle = W - \sum_{i=1}^{k} W_i \cdot \kappa_w(\mathcal{P}_i, \mathcal{P}_i)
\end{aligned}
$$

$$
\min J_W(D) \Longleftrightarrow \max \sum_{i=1}^{k} W_i \cdot \kappa_w(\mathcal{P}_i, \mathcal{P}_i) \tag{12}
$$

C Details of the Unified Framework

C.1 Pointwise Degree-Moment Formulation Used by Algorithm 1

For completeness, the distributional functional in Equation 7 is

$$
g_\lambda(\mathcal{P}_i, \mathcal{P}_D) = \frac{1}{|C_i|} \sum_{x \in C_i} \left(|D| \kappa(\delta(x), \mathcal{P}_D) \right)^\lambda .
$$

It equals 1 at $\lambda = 0$ and $|D|\kappa(\mathcal{P}_i, \mathcal{P}_D)$ at $\lambda = 1$. For $0 < \lambda < 1$, taking the power before averaging is the ordering that yields Equation 8 and Algorithm 1.

Let $h_i = \mathbf{1}_{C_i} / \sqrt{S_\lambda(C_i, D)}$. Because different indicators have disjoint supports,

$$
h_i^\top \mathbf{D}^\lambda h_j = \begin{cases} 1, & i = j, \\ 0, & i \neq j, \end{cases} \qquad h_i^\top K h_i = \frac{S(C_i, C_i)}{S_\lambda(C_i, D)} .
$$

Thus the exact discrete objective is $\mathrm{Tr}(H^\top K H)$ under $H^\top \mathbf{D}^\lambda H = I$ and the discrete support constraints. Dropping only the support constraints and setting $\widetilde{H} = \mathbf{D}^{\lambda/2} H$ gives Equation 9. If all degrees equal d_0, then $S_\lambda(C_i, D) = d_0^\lambda |C_i|$ and $Q_\lambda = d_0^{-\lambda} Q_0$, which is the constant-degree boundary described in Proposition 4.1.

C.2 Power of the Cluster Mean as an Alternative Variant

The order of the two operations can instead be reversed: first aggregate the point-to-distribution similarity over $\mathcal{P}_i$, and then apply the power. This gives

$$
\overline{g}_\lambda(\mathcal{P}_i, \mathcal{P}_D) = \left(|D| \kappa(\mathcal{P}_i, \mathcal{P}_D) \right)^\lambda = \left(\frac{S(C_i, D)}{|C_i|} \right)^\lambda
$$

and, with the same $f(a, b) = a/b$, the alternative objective

$$
\overline{Q}_\lambda(\mathcal{C}) = \sum_{i=1}^{k} \frac{S(C_i, C_i)}{|C_i|^{1-\lambda} S(C_i, D)^\lambda} .
$$

This formulation has the same endpoints as Equation 8: $\overline{Q}_0 = Q_0$ is the kernel k-means association objective and $\overline{Q}_1 = Q_1$ is normalized association. For $0 < \lambda < 1$, however, concavity gives

$$\frac{1}{|C_i|}\sum_{x\in C_i} S(x,D)^\lambda \le \left(\frac{S(C_i,D)}{|C_i|}\right)^\lambda,$$

with equality for a cluster of constant degree. Thus the two distributional orderings define distinct interior objectives whenever within-cluster degrees are heterogeneous.

Unlike the denominator in Equation 8, $m_\lambda(C) = |C|^{1-\lambda}S(C,D)^\lambda$ is not additive over points. Indeed, if a fixed diagonal matrix M satisfied $m_\lambda(C) = \mathbf{1}_C^\top M \mathbf{1}_C$ for every cluster, singleton clusters would require $M_{xx} = S(x,D)^\lambda$. A two-point cluster would then require

$$S(x,D)^\lambda + S(y,D)^\lambda = 2^{1-\lambda}\big(S(x,D)+S(y,D)\big)^\lambda,$$

which is false for $0 < \lambda < 1$ and unequal positive degrees by strict concavity. Consequently, this variant is cluster-coupled and does not reduce to the partition-independent eigendecomposition in Algorithm 1.

It can nevertheless be optimized directly over partitions. Define $a_r = S(C_r, C_r)$, $n_r = |C_r|$, $v_r = S(C_r, D)$, and $F_\lambda(a,n,v) = a/(n^{1-\lambda}v^\lambda)$. For a candidate move $x : C_r \to C_s$ with $n_r > 1$, the affected statistics are

$$n_r^- = n_r - 1, \quad v_r^- = v_r - S(x,D), \quad a_r^- = a_r - 2\sum_{y\in C_r} K_{xy} + K_{xx},$$

$$n_s^+ = n_s + 1, \quad v_s^+ = v_s + S(x,D), \quad a_s^+ = a_s + 2\sum_{y\in C_s} K_{xy} + K_{xx},$$

so its exact objective gain is

$$\Delta_{x:r\to s} = F_\lambda(a_r^-, n_r^-, v_r^-) + F_\lambda(a_s^+, n_s^+, v_s^+) - F_\lambda(a_r, n_r, v_r) - F_\lambda(a_s, n_s, v_s).$$

Algorithm 2 Discrete Refinement for the Cluster-Mean-Powered Variant

Require: Symmetric K with positive degrees, k, $\lambda \in [0,1]$, and a nonempty initial partition $\mathcal{C}$.
Ensure: A partition locally optimal for $\overline{Q}_\lambda$ under admissible single-point moves.

```
Compute a_r, n_r, and v_r for every cluster C_r.
repeat
    Evaluate Δ_{x:r→s} for every x ∈ C_r, s ≠ r, and n_r > 1.
    Let (x*, r*, s*) attain the largest gain Δ*.
    if Δ* > 0 then
        Move x* from C_{r*} to C_{s*} and update the affected statistics.
    end if
until Δ* ≤ 0
return C.
```

Every accepted move strictly increases $\overline{Q}_\lambda$; because there are finitely many partitions, Algorithm 2 terminates at a one-point local optimum. Algorithm 1 may be used for initialization, but the reported experiments use the pointwise degree-moment objective in Equation 8 and Algorithm 1, not this alternative variant.

D EXPERIMENTS SETTING

The Characteristics of the datasets used are shown in Table 3.

The two synthetic datasets are shown in Figure 10.

We use six benchmark datasets: landsat Wang et al. (2023), spambase Hopkins et al. (1999), waveform3 Wang et al. (2023), pendigits Alpaydin & Alimoglu (1996), usps Cai et al. (2010), letters Frey & Slate (1991).

Gaussian kernel was employed in our experiments, we set $\gamma = \frac{1}{d_u^2}$, where $d_u = \frac{2}{n(n-1)}\sum_{i<j}\|x_i - x_j\|$.

Table 3: Characteristics of the datasets used

	Datasets	#points	#dimensions	k
Synthetic	OUTLIER	1000	2	2
	LOW-HIGH	550	2	2
Benchmark	landsat	2000	36	6
	spambase	4601	57	2
	waveform3	5000	21	3
	pendigits	10992	16	10
	usps	11000	256	10
	letters	20000	16	26

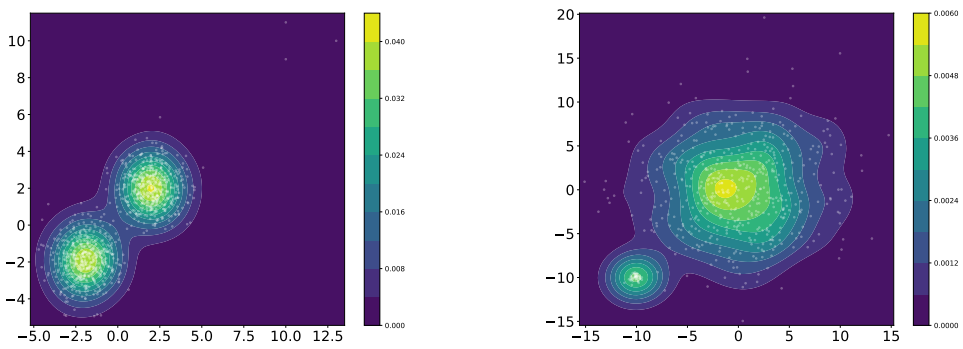

Figure 10: Two synthetic datasets (OUT- LIER (left) and LOW-HIGH (right)).

E LIMITATIONS

Exact dense RBF construction requires $O(|D|^2)$ storage and computation, and finite-feature approximations introduce approximation error. Graph construction and kernel bandwidth can have a larger empirical effect than λ. The eigensolution exactly optimizes the trace-relaxed problem, while row normalization and k-means are heuristic rounding. No universal label-free rule for selecting λ is claimed; this is a separate model-selection question, like choosing bandwidth or graph sparsity. Finally, no-self-loop kNN affinities are not guaranteed PSD, so the kNN results below are graph-objective sensitivity tests rather than KME evidence.

F ADDITIONAL EXPERIMENTAL RESULTS

Protocol. The additional studies use the same input features and Gaussian RBF weights as the main experiments. Dense affinities include self-loops. The fixed-kNN study uses 10 or 20 neighbors, union/max symmetrization, and no self-loops. We evaluate $\lambda = 0, .05, \ldots, 1$, normalize the spectral embedding row-wise, and apply k-means++ with multiple internal restarts. Reported grid values are means over five independent roundings; the maximum standard deviations are .009 NMI and .018 ARI. NMI and ARI use the reference labels only for retrospective evaluation. Since no-self-loop kNN affinities are not guaranteed PSD, those results test the fixed-graph objective rather than the literal RKHS/KME interpretation.

Interior values on fixed dense PSD kernels. Holding the dense RBF affinity fixed, an interior value simultaneously exceeds both endpoints in both metrics on four of the six benchmarks (Table 4). The remaining two datasets do not show such a joint improvement, so the result establishes nonredundancy rather than universal superiority.

Table 4: Interior evidence on fixed dense PSD RBF kernels. Each score is NMI/ARI.

Data	$\lambda = 0$	λ_{int}	Interior	$\lambda = 1$
Spambase	.090/.145	.10	**.092/.148**	.000/.000
Pendigits	.691/.590	.80	**.696/.593**	.695/.591
USPS	.457/.316	.95	**.465/.324**	.464/.323
Letters	.403/.181	.05	**.409/.185**	.397/.163

Matched-input baselines. Table 5 separates degree normalization from other modeling choices. Sparse graph construction is particularly influential on Pendigits and USPS, whereas additive regularization can be competitive but does not reproduce the multiplicative denominator $\sum_{x \in C} S(x, D)^{\lambda}$.

Table 5: Matched-input contextual baselines. Each entry reports NMI/ARI; regularized Ncut uses $D + \tau I$.

Data	Iterative KKM	10NN-SC	20NN-SC	$\tau = .1\bar{d}$	$\tau = \bar{d}$
Landsat	.623/.523	.634/.493	.623/.478	.625/.495	.616/.501
Spambase	.072/.118	.000/.000	.000/.000	.067/.108	.087/.142
Waveform	.366/.255	.370/.251	.370/.252	.363/.249	.366/.250
Pendigits	.700/.590	.842/.773	.844/.779	.695/.592	.695/.594
USPS	.451/.296	.646/.491	.612/.457	.465/.325	.462/.321
Letters	.369/.148	.455/.121	.443/.125	.400/.173	.396/.163

Fixed-graph sensitivity. When only λ is varied on a fixed sparse graph, an interior value improves both metrics over both endpoints in six of the twelve dataset/neighborhood combinations. Interior settings yield non-negligible joint gains in five settings (Table 6); the remaining strict gain on Waveform–20NN is below .001 in both metrics.

Table 6: Fixed-kNN cases with a visible interior improvement. Each score is NMI/ARI.

Graph	$\lambda = 0$	λ_{int}	Interior	$\lambda = 1$
Landsat–20NN	.502/.412	.85	**.627/.502**	.623/.478
Letters–10NN	.433/.182	.30	**.454/.190**	.441/.127
Letters–20NN	.460/.171	.60	**.496/.200**	.442/.124
Spambase–10NN	.011/.026	.90	**.054/.094**	.000/.000
Spambase–20NN	.018/.043	.40	**.024/.051**	.000/.000

The improving interior values include .30, .40, and .60, showing that the effect does not merely approach the Ncut endpoint.

Bandwidth sensitivity. We repeated $\lambda \in \{0, .5, 1\}$ at $.5\gamma, \gamma, 2\gamma$. The Spambase collapse at $\lambda = 1$ persists (NMI $\leq .002$), while $\lambda = .5$ exceeds both endpoints on Pendigits at $.5\gamma$ (.668/.565) and Letters at 2γ (.396/.175). Thus bandwidth and degree normalization are distinct modeling axes.

Alternative solvers and discrete refinement. At the $\lambda = 1$ endpoint, Table 7 compares RatioCut, the standard Ncut relaxation with Ng–Jordan–Weiss rounding, and Dhillon's weighted kernel k-means on the same dense PSD RBF affinity. Weighted kernel k-means is initialized by the Ncut partition.

Table 7: Alternative solvers on the same dense affinity. Solver columns report NMI/ARI; the last column gives NAssoc before and after weighted-kernel refinement.

Data	RatioCut	Ncut–NJW	Weighted KKM	NAssoc init.→final
Landsat	.170/.045	.624/.492	.614/.496	$2.071 \to 2.092$
Spambase	.001/.000	.000/.000	.000/.000	$2.000 \to 2.000$
Waveform	.379/.368	.363/.249	.357/.250	$1.452 \to 1.455$
Pendigits	.002/.000	.695/.591	.700/.582	$1.922 \to 1.938$
USPS	.002/.000	.464/.323	.450/.298	$1.298 \to 1.313$
Letters	.003/.000	.397/.163	.357/.122	$1.787 \to 1.830$

The weighted-kernel refinement never decreases the exact NAssoc objective and converges within 44 updates. External scores may move in either direction because spectral rounding and discrete local refinement are different optimization stages for the same objective.

Nyström approximation. Table 8 summarizes the approximation–efficiency trade-off at $\lambda = .5$. Reported runtimes include feature or affinity construction, eigensolution, and rounding.

Table 8: Nyström approximation at $\lambda = .5$. Scores are NMI/ARI; errors are relative kernel errors (%) at ranks 128/256/512; time and storage compare dense computation with rank 512.

Data	Error 128/256/512	Dense	Rank-512	Time (s)	Storage (MiB)
Pendigits	$.86/.29/.10$	.694/.593	.694/.593	$1.65/.72$	$461/21$
USPS	$4.09/2.45/1.41$	.461/.320	.458/.318	$2.01/.97$	$462/21$
Letters	$1.64/.77/.34$	.405/.176	.400/.176	$6.92/1.74$	$1526/39$

The error decreases with rank. At rank 512, NMI/ARI remains within .005 of the dense result, while runtime is reduced by 2.1–$4.0\times$ and affinity storage by more than $20\times$. All timing comparisons use the same recorded Windows 10 software environment and are interpreted only within-instance; no hardware-independent speed ranking is claimed.