entropy

MDPI

Article

Unsupervised Feature Weighting Based on Graph Heat Kernel and Normalized Hilbert–Schmidt Independence Criterion

Long Zou, Tinghua Wang * and Yuhong Chen

Key Laboratory of Data Science and Artificial Intelligence of Jiangxi Education Institutes, Gannan Normal University, Ganzhou 341000, China; 1240741007@gnnu.edu.cn (L.Z.); 1240740013@gnnu.edu.cn (Y.C.)
* Correspondence: wangtinghua@gnnu.edu.cn

Abstract

Unsupervised feature weighting relies on sample structure to assess feature importance. However, irrelevant features and noise can distort the structural reference, while feature redundancy can cause repeated weighting of similar information. We propose an unsupervised feature-weighting method based on a graph heat kernel and the normalized Hilbert–Schmidt independence criterion (nHSIC), termed GHK-nHSIC-FW, which constructs a label-free structural target using heat diffusion on a nearest-neighbor graph. The method uses nHSIC to measure feature–target alignment and feature redundancy and learns continuous weights with convex quadratic optimization under nonnegative simplex constraints to adjust feature contributions to the weighted Euclidean distance. Experiments on eight public datasets demonstrate that our proposed method can achieve the highest average clustering accuracy (ACC) and normalized mutual information (NMI) among the compared methods, while ablation and subsampling analyses assess component effectiveness and feature importance stability. Synthetic experiments with known informative features examine sample contamination, irrelevant feature injection, correlated feature copies, and diffusion-scale effects under graph perturbations. The results show strong relative performance under sample and irrelevant feature contamination and demonstrate how redundancy control helps retain complementary informative contributions. These results support graph-guided, redundancy-aware feature weighting for unsupervised clustering.

Keywords: unsupervised feature weighting; graph heat kernel; normalized Hilbert–Schmidt independence criterion (nHSIC); feature redundancy; simplex optimization; clustering

1. Introduction

Multivariate representations describe samples from complementary perspectives, but adding variables does not necessarily add useful information. Irrelevant or noisy features can distort Euclidean distances, whereas redundant features may cause the same underlying structure to be counted repeatedly. For clustering algorithms that rely on distances or similarities, both effects can alter neighborhood relations and cluster boundaries. Feature selection performs discrete dimensionality reduction by retaining a subset, but it generally requires a predefined feature count or threshold. Small score variations near the threshold can change the selected subset, and weak yet complementary variables may be removed [1–3]. Feature weighting instead learns continuous scales and therefore adjusts the contribution of each coordinate to downstream distance computation at a finer resolution [4,5]. Embedded approaches such as weighted K-means [6–10] update feature weights jointly with cluster assignments, coupling weight learning to the clustering objective and the evolving partition.

Check for updates

Academic Editor: António Lopes

Received: 10 October 2026
Revised: 9 November 2026
Accepted: 14 November 2026
Published: 16 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

Filter-based unsupervised feature weighting learns weights independently of downstream cluster assignments and requires a data-derived reference for assessing feature relevance without labels. Graph-based methods build on local manifold assumptions. Laplacian Score [11], SPEC [12], and MCFS [13] evaluate features using graph smoothness, spectral filtering, and multi-cluster spectral embedding, respectively, while RUFS [14] combines pseudo-label learning with robust sparse projection. Subsequent studies have developed flexible optimal graphs [15], relevance–redundancy criteria [16], structured graph optimization [17], local structure learning with sparse regression [18], adaptive structure learning [19], redundancy-controlled and similarity-based filters [20,21], and fuzzy anchor graphs [22]. These studies identify the reliability of the structural target and the control of redundant information as two central issues. Nevertheless, most existing methods aim to obtain a sparse subset or projection matrix, leaving room for filter-based continuous weighting that learns continuous feature contributions to distance.

Hilbert–Schmidt independence criterion (HSIC)-based feature evaluation provides another route to nonlinear dependence modeling. HSIC measures dependence using the squared norm of a cross-covariance operator in reproducing kernel Hilbert spaces [23]. HSIC Lasso [24,25] approximates a supervised target kernel with a nonnegative sparse combination of centered single-feature kernels; expanding its squared loss reveals both a feature–target dependence reward and a feature–feature redundancy penalty. UHSIC-Lasso [26] replaces the label kernel with a centered Gaussian kernel constructed from the full input, retaining this relevance–redundancy decomposition in the unsupervised setting while using L1 regularization to obtain a sparse selection. This relevance–redundancy formulation provides a foundation for studying alternative label-free targets when full-input similarities are affected by irrelevant, noisy, or repeated variables and for learning continuous feature weights.

The graph heat kernel offers one possible structural target. K-nearest neighbors (KNN) sparsification restricts direct modeling to local neighborhoods and reduces the immediate influence of distant sample pairs. Graph Laplacians and spectral embedding characterize smooth and oscillatory modes on graphs [27–29], whereas diffusion kernels [30] and diffusion maps [31] integrate one-step adjacency into multistep connectivity. From a graph signal perspective, heat diffusion attenuates high-frequency components more strongly, consistent with multiscale localization in spectral graph analysis [32]. Provided that the initial neighborhoods are reasonably reliable, a graph heat kernel therefore supplies a label-free target distinct from a dense full-input kernel and may reduce sensitivity to unstable global similarities. It does not directly clean the original samples and cannot automatically correct incorrectly constructed edges.

Motivated by the above analyses, we propose an unsupervised feature-weighting method based on a graph heat kernel and the normalized HSIC (nHSIC), termed GHK-nHSIC-FW, which constructs a label-free structural target using heat diffusion on a nearest-neighbor graph. The main contributions are as follows.

(1) We formulate a filter-based unsupervised feature-weighting framework. The learned weights control the contributions of individual features to the weighted Euclidean distance, while single-feature kernels represent the sample structures induced by individual coordinates. Weight learning is decoupled from downstream clustering algorithms such as K-means.

(2) We construct a graph heat kernel target and use nHSIC to obtain scale-comparable structural evidence. A union KNN graph, a normalized graph Laplacian, and heat diffusion jointly provide a multistep diffusion target. The nHSIC between each single-feature kernel and the graph heat kernel evaluates the alignment of that coordinate with the target structure.

(3) Building on the relevance–redundancy principle of HSIC Lasso and UHSIC-Lasso, we embed the nHSIC redundancy matrix into a nonnegative simplex-constrained quadratic program to learn continuous weights. The diagonal terms and an additional diagonal adjustment discourage excessive concentration, while the off-diagonal terms constrain highly overlapping features from simultaneously receiving large weights. Strict convexity on the simplex gives a unique optimum under the stated conditions.

The remainder of this paper is organized as follows. Section 2 reviews unsupervised feature selection and weighting, graph-based learning, and HSIC/nHSIC and positions the proposed method. Section 3 presents the target construction, optimization model, and complexity of GHK-nHSIC-FW. Section 4 reports experiments on public and synthetic data, together with ablation, statistical, and stability analyses. Section 5 concludes the paper and outlines future work.

2. Research Background

2.1. Unsupervised Feature Selection and Feature Weighting

Unsupervised feature selection aims to identify variables that preserve latent structure without class labels and to produce a compact subset. Surveys commonly organize existing approaches into filter scoring, spectral learning, sparse regression, matrix factorization, clustering-driven methods, and joint structure learning [1–3]. Laplacian Score [11] evaluates local smoothness, SPEC [12] unifies spectral criteria using graph filtering, MCFS [13] uses sparse regression to reconstruct a multi-cluster spectral embedding, and RUFS [14] jointly learns robust representations, pseudo-labels, and sparse features. DGUFS [33] uses an L2,0 equality constraint to control the selected feature count directly and couples the original data, pseudo-cluster labels, and selected features in a joint selection-clustering framework. Later methods have emphasized graph reliability and redundancy control using structured graph optimization [17], adaptive graph learning [19], feature similarity filters [21], and controlled redundancy [20], while fuzzy anchor graphs improve robustness under noisy conditions [22]. NDFS [34] jointly learns nonnegative spectral cluster indicators and a row-sparse feature selection matrix. Their outputs are typically feature rankings, sparse coefficients, or selected subsets.

Feature weighting instead focuses on the relative contribution of each variable to similarity or distance computation and does not necessarily return a subset of fixed size. Reviews classify weighting methods as filter, wrapper, or embedded approaches according to the source of the weights and the downstream task; weighted K-means is a major unsupervised branch [4,5,35]. Early methods incorporated variable weights into within-cluster dispersion and controlled their distributions using exponent or entropy regularization [6,9,10]. More recently, LLSRFS [18] has combined local structure learning, sparse regression, and exponential feature weighting and ultimately ranks the learned weights to select a Top-k subset. Although its intermediate weights are continuous, LLSRFS remains an embedded feature selection method because the weights are optimized jointly with its representation and graph variables and are used for final subset selection. By contrast, we adopt a filter setting in which continuous weights are learned independently of downstream cluster assignments and applied before clustering. Importantly, a filter score is not equivalent to the predictive or clustering performance of a specific downstream model, so task utility must still be validated independently [36].

2.2. Graph-Based Unsupervised Feature Learning

Graph-based methods represent samples as nodes, connect locally similar samples, and use graph Laplacians to encode data geometry. Classical methods such as Laplacian

Score [11], SPEC [12], and MCFS [13] evaluate features directly using graph smoothness or spectral embeddings. Structured graph optimization [17], flexible optimal graphs [15], and adaptive structure learning [19] jointly optimize similarities and feature variables to reduce the bias of a fixed graph. Robust anchor graph methods reduce graph construction cost while improving tolerance to noise [22]. More generally, spectral embedding theory relates the low-frequency subspace of a graph Laplacian to manifold coordinates [29].

Diffusion methods integrate local edges into multistep connectivity. Kernels and regularization on graphs connect spectral functions of a graph Laplacian with kernel methods [28], while diffusion kernels [30] and diffusion maps [31] describe multiscale geometry through discrete heat propagation and stochastic diffusion, respectively. Spectral graph wavelets further show that selecting graph frequencies controls both smoothness and localization [32]. Graph heat diffusion can therefore supply a label-free structural target for feature evaluation that reflects local connectivity and multistep relationships.

2.3. HSIC and nHSIC for Feature Learning

HSIC is a nonparametric dependence measure defined in reproducing kernel Hilbert spaces. It quantifies dependence through the squared Hilbert–Schmidt norm of the cross-covariance operator between two kernel representations; with characteristic kernels and the required regularity conditions, a population HSIC of zero characterizes independence [23]. Empirical HSIC is computed from the inner product of centered Gram matrices, avoids explicit joint density estimation, and captures nonlinear relations through kernels. In practice, biased and unbiased estimators must be distinguished, and kernel type, bandwidth, and sample size jointly affect the statistic. Direct Gram matrix construction also incurs quadratic sample storage and computation, as reviewed in [37].

HSIC and mutual information both measure statistical dependence, but they are not interchangeable. Mutual information feature selection has a clear information-theoretic interpretation; for continuous variables, however, it often requires density, entropy, or nearest-neighbor estimation and is sensitive to estimator bias and parameter choices [38]. Squared-loss mutual information offers another directly estimable criterion based on density ratios and squared loss [39]. HSIC avoids explicit density estimation using kernel means and covariance operators, at the cost of dependence on the kernel representation and bandwidth. Mutual information, squared-loss mutual information, and HSIC should therefore be viewed as distinct dependence evaluation routes rather than substitutes identified by a common numerical scale.

Early HSIC-based feature selection primarily measured dependence between candidate feature sets and supervised labels. BAHSIC [40] uses HSIC to guide backward elimination of features with respect to supervised labels. Song et al. [41] subsequently developed a broader dependence maximization framework for feature selection. This search-based route extends feature evaluation from marginal linear association to nonlinear dependence between a feature set and a target, but it requires a retained feature budget and repeatedly recomputes candidate sets. It therefore establishes target-guided subset search rather than continuous weights for distance computation.

HSIC Lasso [24] reformulates subset search as convex sparse kernel fitting. Each input dimension is represented by a centered feature kernel, a nonnegative linear combination of these kernels approximates a target kernel constructed from the outputs, and an L1 penalty drives unimportant coefficients to zero. Expanding the squared fitting loss reveals a feature–target dependence reward, pairwise overlap penalties among feature kernels, and the L1 sparsity term. HSIC Lasso can thus be interpreted as a maximum relevance/minimum redundancy framework in kernel space rather than independent HSIC scoring of each feature. Its convexity and statistical interpretation are important advan-

tages, whereas kernel bandwidth, regularization, and quadratic Gram matrix costs remain practical limitations [25].

Normalized HSIC (nHSIC) scales the inner product between centered kernels by their Frobenius norms. Under the corresponding centering and normalization definitions, nHSIC has the same geometric form as centered kernel alignment (CKA) [25,42]. Normalization removes the direct influence of overall kernel magnitude and places the similarity between a single-feature kernel and a target, as well as that between two single-feature kernels, on a common scale. The normalized similarity remains dependent on the kernel choice and bandwidth.

In the absence of labels, the key transition is to construct a target from the data themselves. Liaghat and Mansoori proposed SP-BAHSIC and SP-FOHSIC [43], which use backward elimination and forward selection, respectively, to preserve the sample similarity structure of the full feature set using $HSIC_2$. Extensions employ feature clustering or Plus-l-Minus-r search to reduce cost and local-search risk. These methods re-evaluate how well a candidate subset preserves full-input similarity at each step, but still require a target feature count and output a discrete subset. Unlike these sequential search methods, UHSIC-Lasso [26] replaces the centered label target of HSIC Lasso with a centered Gaussian kernel constructed from the full input and fits this target using nonnegative L1 optimization. After expansion, the first term rewards HSIC between each feature kernel and the full-input kernel, the second penalizes HSIC between feature kernels, and the L1 term induces sparsity. Although the original study normalizes the nonnegative coefficients after optimization, its objective is still to remove irrelevant or redundant variables and obtain a subset, not to learn continuous metric weights. A unified view further shows that supervised, unsupervised, and semi-supervised feature selection can be interpreted consistently using target kernel construction and relevance–redundancy decomposition [26]. Chen et al. [44] proposed U-HSIC-ENet, which combines self-supervised kernel alignment, an explicit off-diagonal redundancy penalty, and elastic net-type regularization for unsupervised feature selection. These studies provide the most direct foundation for our method. HSIC and related kernel alignment criteria have also been applied to kernel learning for supervised prediction [25,37], where label-derived targets distinguish these applications from the unsupervised feature-weighting task considered here.

Overall, the HSIC family offers clear advantages for nonlinear dependence representation and unified relevance–redundancy modeling, but three limitations are directly relevant here. First, an unsupervised reference is often constructed from the full input or its kernel and may inherit similarities already distorted by irrelevant, noisy, or repeated variables. Second, BAHSIC-style search and L1-sparse optimization primarily select subsets and require a feature budget or regularization strength to control sparsity. Third, unnormalized HSIC varies with kernel magnitude across features, whereas normalization does not remove kernel parameter sensitivity or quadratic sample complexity [25,26,37,43]. These limitations motivate the structural target and weighting formulation studied here.

Building on these foundations, the proposed framework combines three choices: replace the full-input Gaussian target with heat diffusion on a union KNN graph to obtain a locally constrained label-free target; use nHSIC to place single-feature–target alignment and single-feature–single-feature interactions on a common scale; and learn continuous weights with a redundancy matrix and nonnegative simplex-constrained quadratic optimization rather than obtaining a subset of predefined size using backward elimination or L1 sparsity.

2.4. Positioning of the Proposed Method

We position GHK-nHSIC-FW as a filter-based unsupervised feature-weighting method. The framework develops the relevance–redundancy principle using a graph-diffusion target

and continuous weighting constraints. The graph heat kernel provides a structural target shaped by local-neighborhood constraints and spectral diffusion. nHSIC places the relevance vector and redundancy matrix on a consistent normalized scale, and a nonnegative simplex-constrained convex quadratic program produces interpretable continuous feature weights. The weights are learned once before clustering, require no predefined number of retained features, and do not depend on the current partition produced by downstream K-means. Masoomi et al. [45] investigated feature redundancy from an instance-wise grouping perspective using information-theoretic criteria for classification. In contrast, GHK-nHSIC-FW evaluates feature redundancy using pairwise nHSIC and learns a global continuous weight vector from unlabeled data.

3. The GHK-nHSIC-FW Method

Standardized data provide a local graph for the diffusion target and a separate kernel representation for each feature. nHSIC evaluates feature–target relevance and feature redundancy, and a simplex-constrained quadratic program learns continuous weights. For fixed hyperparameters, this procedure is independent of the downstream K-means partition.

3.1. Feature Weights and Weighted Representation

Let $\mathbf{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]^\top \in \mathbb{R}^{n \times d}$ contain n samples and d features. After feature-wise standardization, denote the data by $\mathbf{X}_{\text{std}}$. We learn a nonnegative feature weight vector on the simplex as follows:

$$\Delta_d = \{\boldsymbol{w} \in \mathbb{R}^d : w_j \geq 0, \mathbf{1}^\top \boldsymbol{w} = 1\}. \tag{1}$$

The weighted representation and its squared Euclidean distance are expressed as follows:

$$\mathbf{X}^{(w)} = \mathbf{X}_{\text{std}} \text{diag}\left(\sqrt{\boldsymbol{w}}\right), \tag{2}$$

$$d^2_{\boldsymbol{w}}(\boldsymbol{x}_a, \boldsymbol{x}_b) = \sum_{j=1}^{d} w_j \left(\boldsymbol{x}^{\text{std}}_{aj} - \boldsymbol{x}^{\text{std}}_{bj}\right)^2. \tag{3}$$

Each weight controls a standardized coordinate's contribution to distance. The simplex constraint fixes the overall scale and makes relative contributions comparable. The feasible set includes zero weights. For numerical evaluation of the proposed method and its variant without redundancy, the coordinate scaling uses $\sqrt{w_j + 10^{-10}}$.

Single-feature kernels provide structural evidence for learning these weights. They describe the sample relationships induced by each coordinate; the optimization rewards agreement with the diffusion target while penalizing redundant contributions. The final output remains a vector of feature weights, which can be applied before downstream clustering.

3.2. Graph Heat Kernel Target

We standardize the feature columns and remove zero-variance columns. Let $\mathcal{N}_k(\boldsymbol{x}_a)$ be the set of the k nearest neighbors of sample $\boldsymbol{x}_a$ in the standardized space. Union symmetrization retains an edge if either sample belongs to the other's neighborhood:

$$A_{ab} = \begin{cases} 1, & \boldsymbol{x}_b \in \mathcal{N}_k(\boldsymbol{x}_a) \text{ or } \boldsymbol{x}_a \in \mathcal{N}_k(\boldsymbol{x}_b), \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

For $a \neq b$, the weighted adjacency matrix is expressed as follows:

$$W_{ab} = A_{ab} \exp\left(-\frac{\| \boldsymbol{x}^{\text{std}}_a - \boldsymbol{x}^{\text{std}}_b \|_2^2}{2\sigma_g^2}\right), \qquad W_{aa} = 1. \tag{5}$$

The edge weight bandwidth σ_g is the median of strictly positive Euclidean distances between the endpoints of edges in the union KNN graph. The diagonal entries of the weighted

adjacency matrix equal one. Thus, σ_g controls the local edge weights, whereas τ controls subsequent heat diffusion.

With $\mathbf{D}_{aa} = \sum_b W_{ab}$, the symmetric normalized Laplacian and graph heat kernel are expressed as follows:

$$\mathbf{L}_{\mathrm{sym}} = \mathbf{I} - \mathbf{D}^{-1/2} W \mathbf{D}^{-1/2}, \qquad \mathbf{K}_{\mathrm{G}} = \exp(-\tau \mathbf{L}_{\mathrm{sym}}). \tag{6}$$

Here, $\tau > 0$ is the diffusion time parameter controlling the diffusion scale. For $\mathbf{L}_{\mathrm{sym}} = U \mathrm{diag}(\mu_1,\ldots,\mu_n) U^{\top}$, this gives the following:

$$\mathbf{K}_{\mathrm{G}} = U \mathrm{diag}(e^{-\tau\mu_1},\ldots,e^{-\tau\mu_n}) U^{\top}. \tag{7}$$

The heat kernel applies the spectral response $e^{-\tau\mu}$ to each graph mode. This positive response yields a positive semidefinite kernel and attenuates larger graph frequencies more strongly [27–32]. For a graph signal $z = \sum_i \alpha_i u_i$, diffusion gives $K_{\mathrm{G}} z = \sum_i e^{-\tau\mu_i} \alpha_i u_i$, explicitly connecting the target to multistep graph smoothing. The benefit depends on whether the initial graph captures useful neighborhoods: attenuation is defined relative to that graph and does not guarantee correction of erroneous edges. The experiments examine this dependence through irrelevant feature injection and shared graph perturbations.

3.3. nHSIC-Based Feature Relevance

For feature j, we construct a one-dimensional Gaussian kernel as follows:

$$\left(\mathbf{K}_j\right)_{ab} = \exp\left[-\frac{\left(x_{aj}^{\mathrm{std}} - x_{bj}^{\mathrm{std}}\right)^2}{2\sigma_j^2}\right]. \tag{8}$$

The feature bandwidth σ_j is determined by the median of positive pairwise distances on that feature. Define $\mathbf{H} = \mathbf{I} - 11^{\top}/n$ and the centered kernel $\widetilde{\mathbf{K}} = \mathbf{HKH}$. The biased empirical HSIC estimator and its normalized form are expressed as follows:

$$\mathrm{HSIC}(\mathbf{K},\mathbf{L}) = \frac{\mathrm{tr}(\mathbf{KHLH})}{(n-1)^2}, \tag{9}$$

$$\mathrm{nHSIC}(\mathbf{K},\mathbf{L}) = \frac{\left\langle \widetilde{\mathbf{K}}, \widetilde{\mathbf{L}} \right\rangle_{\mathrm{F}}}{\| \widetilde{\mathbf{K}} \|_{\mathrm{F}} \| \widetilde{\mathbf{L}} \|_{\mathrm{F}}}. \tag{10}$$

Small denominators are protected numerically. For nondegenerate centered kernels, this normalized expression coincides with centered kernel alignment [42]. It compares kernel structures on a common scale while retaining dependence on the chosen kernel and bandwidth. The relevance score is expressed as follows:

$$g_j = \mathrm{nHSIC}\left(\mathbf{K}_j, \mathbf{K}_{\mathrm{G}}\right), \qquad g = \left[g_1,\ldots,g_d\right]^{\top}. \tag{11}$$

A larger score indicates stronger agreement between the structure induced by feature j and the diffusion target. This score provides a label-free measure of structural relevance.

3.4. Redundancy Matrix

Features that induce similar sample relationships may jointly amplify a repeated structural direction. We evaluate their redundancy using the same dependence criterion:

$$B_{ij} = \mathrm{nHSIC}(\mathbf{K}_i,\mathbf{K}_j), \qquad 1 \le i,j \le d. \tag{12}$$

Let $\overline{\mathbf{K}}_j = \mathbf{HK}_j\mathbf{H} / \| \mathbf{HK}_j\mathbf{H} \|_{\mathrm{F}}$ for a nondegenerate feature kernel. Then, $B_{ij} = \left\langle \overline{\mathbf{K}}_i, \overline{\mathbf{K}}_j \right\rangle_{\mathrm{F}}$. For any $v \in \mathbb{R}^d$, the following is expressed:

$$v^{\top}\mathbf{B}v = \| \sum_{j=1}^{d} v_j \overline{\mathbf{K}}_j \|_{\mathrm{F}}^2 \ge 0. \tag{13}$$

The nHSIC redundancy matrix is therefore positive semidefinite. Its off-diagonal entries measure structural overlap between different features, while its diagonal entries equal one for nondegenerate kernels. The corresponding quadratic penalty discourages highly similar features from simultaneously receiving large weights. The diagonal also penalizes concentrated weight allocations.

3.5. Simplex-Constrained Weight Learning

Let $g_{\min} = \min_j g_j$, $b_{\min} = \min_{i,j} B_{ij}$, $s_g = \max_j g_j - g_{\min}$, and $s_{\mathbf{B}} = \max_{i,j} B_{ij} - b_{\min}$. The optimization coefficients are $\boldsymbol{h} = (\boldsymbol{g} - g_{\min}\mathbf{1})/s_g$ and $\mathbf{R} = (\mathbf{B} - b_{\min}11^\top)/s_{\mathbf{B}}$; a zero range gives a zero vector or matrix. We learn the feature weights by solving the following:

$$\min_{\boldsymbol{w}\in\Delta_d} J(\boldsymbol{w}) = -\boldsymbol{h}^\top \boldsymbol{w} + \boldsymbol{w}^\top \mathbf{R}\boldsymbol{w} + \lambda \parallel \boldsymbol{w} \parallel_2^2, \qquad \lambda > 0. \tag{14}$$

The linear term rewards agreement with the graph target. The redundancy term limits the simultaneous weighting of overlapping feature structures, and the additional squared-norm term discourages excessive concentration. Its standard quadratic-program form is expressed as follows:

$$\min_{\boldsymbol{w}\geq 0,\, 1^\top \boldsymbol{w}=1} \frac{1}{2}\boldsymbol{w}^\top \mathbf{Q}\boldsymbol{w} + \boldsymbol{c}^\top \boldsymbol{w}, \qquad \mathbf{Q} = 2(\mathbf{R} + \lambda \mathbf{I}), \quad \boldsymbol{c} = -\boldsymbol{h}. \tag{15}$$

Proposition 1. *Suppose* $\boldsymbol{v}^\top \mathbf{R}\boldsymbol{v} \geq 0$ *for every* $\boldsymbol{v}$ *satisfying* $\mathbf{1}^\top \boldsymbol{v} = 0$. *For* $\lambda > 0$, *Equation (14) is strictly convex on the simplex and admits a unique minimizer.*

Proof. For any two distinct feasible points, their difference $\boldsymbol{v}$ is nonzero and satisfies $1^\top \boldsymbol{v} = 0$. The second directional derivative of the objective along this feasible direction is $2\boldsymbol{v}^\top \mathbf{R}\boldsymbol{v} + 2\lambda \parallel \boldsymbol{v} \parallel_2^2 > 0$. Thus, the objective is strictly convex on the simplex. Existence follows from continuity on the nonempty compact feasible set, and strict convexity establishes uniqueness. For $s_{\mathbf{B}} > 0$ and $\mathbf{1}^\top \boldsymbol{v} = 0$, the coefficient definition gives $\boldsymbol{v}^\top \mathbf{R}\boldsymbol{v} = \boldsymbol{v}^\top \mathbf{B}\boldsymbol{v}/s_{\mathbf{B}} \geq 0$ in Equation (13); when $s_{\mathbf{B}} = 0$, $\mathbf{R} = 0$. Thus, the stated condition holds. □

On the nonnegative simplex, $\|\boldsymbol{w}\|_1 = 1$, so adding an L1 penalty would only add a constant. The present formulation instead controls the relative magnitude of the weights through quadratic penalties. It produces continuous feature contributions without prescribing a retained feature count, but its optimum can lie on the boundary of the simplex.

3.6. Algorithm and Computational Complexity

Algorithm 1 summarizes GHK-nHSIC-FW, including graph heat kernel construction, feature relevance and redundancy evaluation, and simplex-constrained weight optimization.

Algorithm 1. GHK-nHSIC-FW

Input: unlabeled data X; neighborhood size k**; diffusion parameter** τ**; kernel bandwidth settings; regularization coefficient** $\lambda > 0$.

Output: feature weights $\boldsymbol{w}^\star$ and weighted representation $\mathbf{X}^{(\boldsymbol{w}^\star)}$.

1. Standardize the feature columns and remove zero-variance columns.
2. Construct the union KNN graph and Gaussian edge weights using Equations (4) and (5).
3. Compute the normalized Laplacian and its heat kernel using Equations (6) and (7).
4. Construct and center the single-feature kernels.
5. Evaluate feature relevance and redundancy using Equations (11) and (12), and assemble the optimization coefficients.
6. Solve the simplex-constrained quadratic program in Equation (15).
7. Return $\boldsymbol{w}^\star$ and the weighted representation in Equation (2).

The quadratic program is solved using sequential least-squares programming (SLSQP) in SciPy 1.15.3 (Python 3.10.6), with uniform initialization $w_j = 1/d$, a maximum of 1000 iterations, and a function tolerance of 10^{-12}. Feasibility checks require nonnegative weights and a unit sum within 10^{-6}; unsuccessful or infeasible candidates are excluded from parameter selection. This directly optimizes the constrained objective in Equation (15), rather than alternating weight updates with K-means assignments.

Pairwise distances require $O(n^2d)$ operations, and direct neighbor sorting requires $O(n^2 \log n)$. Exact graph eigendecomposition requires $O(n^3)$ time. Constructing feature kernels and relevance scores costs $O(dn^2)$; computing all feature pair nHSIC scores costs $O(d^2n^2)$. A linear algebra step in the dense quadratic program can cost $O(d^3)$, with total optimization time also depending on the solver and its iterations. The leading construction costs, together with a dense optimization step, are therefore expressed as follows:

$$O(n^2d + n^3 + d^2n^2 + d^3). \tag{16}$$

Caching all feature kernels requires $O(dn^2 + d^2)$ storage. A blockwise or recomputation strategy can reduce kernel storage at the cost of additional computation or data movement. The exact formulation is consequently most appropriate for moderate problem sizes; large-sample and very high-dimensional applications require approximate diffusion and more economical kernel pair computation.

4. Experiments and Results

We evaluate the method on eight public datasets with a unified continuous-weighting protocol, examine component contributions and statistical comparisons, and investigate feature weights and clustering behavior using controlled synthetic perturbations and subsampling stability tests.

4.1. Experimental Settings

Eight benchmark datasets are used: Urban Land Cover (Urban), Dermatology, MONK-3, Zoo, QSAR Biodegradation (QSAR), Glass, Haberman, and Libras [46]. Their sizes and class counts are shown in Table 1. All methods are compared using the same standardized input representation, with the number of clusters set equal to the number of reference classes. Data preparation performs mean imputation for the eight missing Dermatology entries, preserves the available integer encodings of categorical attributes (including MONK-3), and applies feature-wise Z-score standardization. Input columns are checked for zero variance; none of the eight prepared datasets contains a zero-variance feature.

Table 1. Benchmark dataset characteristics.

Dataset	Samples	Features	Classes
Urban	675	147	9
Dermatology	366	34	6
MONK-3	432	6	2
Zoo	101	16	7
QSAR	1055	41	2
Glass	214	9	6
Haberman	306	3	2
Libras	360	90	15

We compare GHK-nHSIC-FW with standard K-means using equal feature weights (KM), Laplacian Score (LS-W) [11], LLSRFS-W [18], MCFS-W [13], UHSIC-Lasso-W [26], NDFS-W [34], and U-HSIC-ENet [44]. The suffix "-W" denotes a continuous-weighting adaptation of a feature-ranking or feature selection method. LLSRFS and the two HSIC

selection baselines use their nonnegative feature weight or coefficient vectors, normalized to a unit sum. All methods use the same weighted Euclidean representation in Equation (2), followed by K-means. In the result tables, UHSIC-W, ENet-W, and GHK-FW abbreviate UHSIC-Lasso-W, U-HSIC-ENet, and GHK-nHSIC-FW, respectively.

For LS-W, the scores are reversed as $q_j = \max_k \mathrm{LS}_k - \mathrm{LS}_j$ and normalized with $w_j = q_j / \sum_k q_k$. Equal scores give uniform weights, while nonfinite scores are invalid. For MCFS-W, $q_i = \max_j |C_{ij}|$, where C contains the sparse regression coefficients for the embedding coordinates. For NDFS-W, $q_i = \|V_{i,:}\|_2$, where V is the learned feature selection matrix. These nonnegative scores are divided by their sum. All-zero or nonfinite MCFS/NDFS scores are treated as invalid; MCFS fits with insufficient effective support are also excluded. For each valid weighting vector, feature scaling uses its square root, so that weights enter squared Euclidean distances linearly.

For GHK-nHSIC-FW, the neighborhood size is fixed at $k = 10$, and $\tau \in \{0.1,0.3,1,2,5,10\}$. The regularization grid is $\lambda = s\,10^{a_\ell}$, where $a_\ell = -3 + 6\ell/9$ for $\ell = 0,\ldots,9$. Thus, the ten multipliers are logarithmically spaced from 10^{-3} to 10^{3}. Define $\mathcal{D} = \{R_{ij} : i < j, R_{ij} > 10^{-10}\}$. The scale is $s = \max\{\mathrm{median}(\mathcal{D}), 10^{-6}\}$ when $\mathcal{D}$ is nonempty, and $s = 1$ otherwise. Table 2 specifies the parameter grids and fixed settings for all methods.

Table 2. Hyperparameter grids and fixed numerical settings.

Method	Parameter Grid	Fixed Settings
KM	None	Equal feature weights; common K-means configuration
LS-W	$k \in \{3,5,10,15,20\}$; bandwidth multiplier $\eta \in \{0.25,0.5,1,2,4\}$; additionally $k = 5$, $\sigma = 1$	$\sigma = \eta$ times the median Euclidean distance on union KNN edges
LLSRFS-W	$\beta \in \{0.1,0.5,1\}$; $\lambda_p, \lambda_w \in \{0.01,0.1,1,10,100\}$	$k = 5$; 100 iterations
MCFS-W	$k \in \{3,5,7,10,15,20,30,40,50,70\}$; internal sparsity budgets from 10% to 90% in 10% steps; duplicate feature counts removed	Binary union KNN graph; trivial embedding vector retained; least-angle regression
UHSIC-Lasso-W	20 geometrically spaced relative penalties r_λ from 0.001 to 0.99	$\lambda = r_\lambda \max(b)$; median Gaussian kernels; nHSIC; Min–Max scaling of b, with Q unchanged
NDFS-W	$\alpha, \beta \in \{0.01,0.1,1,10,100\}$	$k = 5$; $\gamma = 10^9$; ridge 10^{-6}; seed 0; 1000 iterations
U-HSIC-ENet	$\lambda_1 \in \{0.01,0.05,0.1,0.2\}$; $\lambda_2 \in \{0,0.001,0.01,0.05,0.1\}$; $\gamma \in \{0,0.1,0.3,0.5,1\}$	Median Gaussian kernels; nHSIC; no Min–Max scaling of Q or b; 800 iterations
GHK-nHSIC-FW	$\tau \in \{0.1,0.3,1,2,5,10\}$; $\lambda = s\,10^{a_\ell}$, $a_\ell = -3 + 6\ell/9$, $\ell = 0,\ldots,9$	$k = 10$; relevance and redundancy coefficients as in Section 3.5; SLSQP as in Section 3.6

For the HSIC baselines, b and Q denote their relevance vector and quadratic coefficient matrix, respectively. Gaussian bandwidths follow the median distance rule. Geometric spacing includes both endpoints; the proposed method uses the logarithmic grid and scale s defined above. MCFS internal sparsity budgets are distinct from the five retained feature budgets examined in Section 4.2.

For each method, all candidate parameter combinations specified in Table 2 were evaluated on each dataset. The best clustering results, defined by the highest mean clustering accuracy (ACC), are reported for comparison. The corresponding normalized mutual information (NMI) [47] is obtained using the same parameter configuration. Feature learning for each configuration does not use class labels; reference labels are used for the external clustering assessment and to identify the reported best ACC results. K-means uses K-means++ initialization, 20 internal initializations, Lloyd updates, at most 300 iterations,

and a tolerance of 10^{-4}. Public data results are means and sample standard deviations over 30 K-means runs; macro-averages give the eight datasets equal weight.

4.2. Clustering Performance on Public Datasets

Tables 3 and 4 give mean ± standard deviation ACC and NMI across the eight datasets. GHK-nHSIC-FW obtains the highest macro-average ACC (0.7063) and NMI (0.4679) compared with the best baseline macro-averages of 0.6413 and 0.4359, respectively. The method obtains the highest per-dataset average on six of eight datasets for ACC and six of eight for NMI. Alternative methods lead on Zoo and Glass in ACC and on Zoo and Libras in NMI.

Table 3. ACC results (mean ± standard deviation); bold indicates the highest mean for each dataset.

Dataset	KM	LS-W	LLSRFS-W	MCFS-W	UHSIC-W	NDFS-W	ENet-W	GHK-FW
Urban	0.5919 ± 0.0484	0.6386 ± 0.0396	0.6931 ± 0.0048	0.6495 ± 0.0382	0.6137 ± 0.0105	0.6569 ± 0.0351	0.6480 ± 0.0280	**0.7076 ± 0.0059**
Dermatology	0.7364 ± 0.0128	0.8379 ± 0.0611	0.8704 ± 0.0035	0.8792 ± 0.0231	0.7399 ± 0.0011	0.7772 ± 0.0014	0.8597 ± 0.0381	**0.9120 ± 0.0098**
MONK-3	0.5106 ± 0.0353	0.5000 ± 0.0000	0.5228 ± 0.0000	0.5000 ± 0.0000	0.5000 ± 0.0000	0.6389 ± 0.0000	0.5407 ± 0.0000	**0.6407 ± 0.0342**
Zoo	0.8234 ± 0.0227	0.9007 ± 0.0066	0.8564 ± 0.0252	0.8607 ± 0.0025	0.8924 ± 0.0178	0.8845 ± 0.0075	**0.9092 ± 0.0090**	0.9023 ± 0.0096
QSAR	0.6210 ± 0.0008	0.7074 ± 0.0014	0.6398 ± 0.0000	0.6350 ± 0.0000	0.6904 ± 0.0036	0.6427 ± 0.0000	0.6891 ± 0.0000	**0.7198 ± 0.0010**
Glass	0.4512 ± 0.0118	0.4980 ± 0.0223	0.5020 ± 0.0124	**0.5567 ± 0.0170**	0.5466 ± 0.0061	0.5461 ± 0.0053	0.4522 ± 0.0069	0.5347 ± 0.0175
Haberman	0.5205 ± 0.0028	0.5229 ± 0.0000	0.5434 ± 0.0033	0.5194 ± 0.0008	0.5206 ± 0.0039	0.5065 ± 0.0000	0.5261 ± 0.0000	**0.7549 ± 0.0000**
Libras	0.4573 ± 0.0198	0.4535 ± 0.0107	0.4758 ± 0.0150	0.4610 ± 0.0187	0.4202 ± 0.0157	0.4772 ± 0.0164	0.4621 ± 0.0145	**0.4781 ± 0.0117**
Macro-average	**0.5890**	**0.6324**	**0.6380**	**0.6327**	**0.6155**	**0.6413**	**0.6359**	**0.7063**

Note: Bold indicates the highest mean for each dataset and the macro-average.

Table 4. NMI results (mean ± standard deviation); bold indicates the highest mean for each dataset.

Dataset	KM	LS-W	LLSRFS-W	MCFS-W	UHSIC-W	NDFS-W	ENet-W	GHK-FW
Urban	0.5418 ± 0.0156	0.5686 ± 0.0127	0.6194 ± 0.0039	0.5667 ± 0.0068	0.5566 ± 0.0036	0.5722 ± 0.0233	0.5750 ± 0.0085	**0.6268 ± 0.0055**
Dermatology	0.8605 ± 0.0050	0.8631 ± 0.0054	0.8639 ± 0.0075	0.8680 ± 0.0121	0.8609 ± 0.0001	0.8423 ± 0.0026	0.8613 ± 0.0078	**0.8783 ± 0.0168**
MONK-3	0.0039 ± 0.0144	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0566 ± 0.0000	0.0000 ± 0.0000	**0.0620 ± 0.0442**
Zoo	0.8330 ± 0.0262	0.9109 ± 0.0087	0.9152 ± 0.0266	0.8948 ± 0.0033	0.9031 ± 0.0115	0.9043 ± 0.0052	**0.9239 ± 0.0055**	0.9140 ± 0.0098
QSAR	0.1200 ± 0.0006	0.1517 ± 0.0018	0.0144 ± 0.0000	0.1410 ± 0.0000	0.1089 ± 0.0025	0.0973 ± 0.0000	0.1123 ± 0.0000	**0.1703 ± 0.0009**
Glass	0.3071 ± 0.0146	0.3655 ± 0.0216	0.4086 ± 0.0181	0.4145 ± 0.0161	0.4243 ± 0.0067	0.3981 ± 0.0105	0.3447 ± 0.0110	**0.4291 ± 0.0297**
Haberman	0.0009 ± 0.0001	0.0011 ± 0.0000	0.0019 ± 0.0003	0.0009 ± 0.0002	0.0012 ± 0.0002	0.0007 ± 0.0000	0.0010 ± 0.0000	**0.0682 ± 0.0000**
Libras	0.5920 ± 0.0113	0.5841 ± 0.0105	0.6151 ± 0.0089	0.5896 ± 0.0111	0.5279 ± 0.0102	**0.6160 ± 0.0088**	0.5937 ± 0.0098	0.5947 ± 0.0096
Macro-average	**0.4074**	**0.4306**	**0.4298**	**0.4344**	**0.4229**	**0.4359**	**0.4265**	**0.4679**

Note: Bold indicates the highest mean for each dataset and the macro-average.

Figure 1 summarizes the eight dataset-level means for each method. The proposed method has the highest cross-dataset average on both metrics. The strongest individual result nevertheless depends on the dataset. U-HSIC-ENet gives higher ACC and NMI on Zoo, MCFS-W gives higher ACC on Glass, and NDFS-W gives higher NMI on Libras.

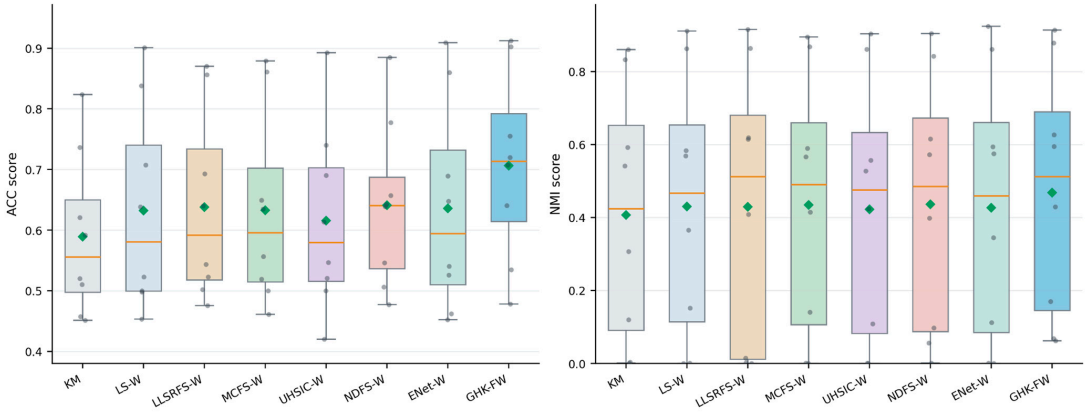


Figure 1. Cross-dataset ACC and NMI performance across eight public datasets. Boxes summarize dataset-level mean scores. Gray dots denote individual datasets, orange lines indicate medians, and green diamonds indicate means.

Figure 2 examines Top-m selection using the feature rankings obtained from the weighted comparison. The five nominal retained feature ratios are $r \in \{0.2,0.4,0.6,0.8,0.9\}$, with $m = \min\{d-1, \max\{1, \text{round}(dr)\}\}$. Rounding follows Python's nearest integer convention, with ties to even. The selected parameter configuration and feature ranking remain fixed across budgets. The top m standardized coordinates are used directly in 30 K-means runs without continuous reweighting. For UHSIC-Lasso and U-HSIC-ENet, all effective nonzero features are retained when their number is below m. Repeated feature counts retain their nominal budget labels and reuse the same subset results; for example, Haberman uses 1, 1, 2, 2, and 2 features. KM and full continuous GHK-nHSIC-FW are horizontal references. This diagnostic measures the effect of truncating the learned rankings and does not retune selection methods separately for each budget.

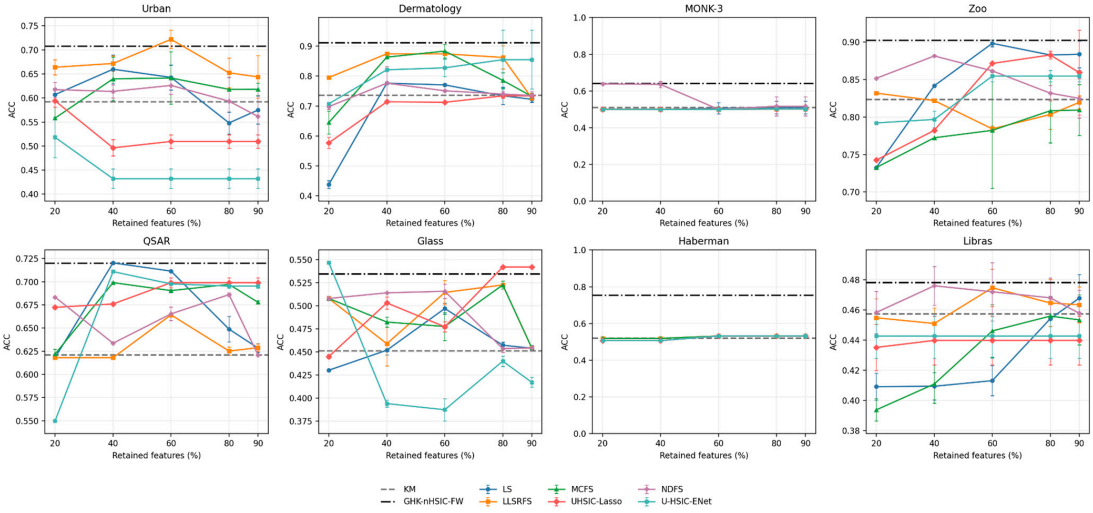


Figure 2. ACC under five nominal retained feature budgets using fixed parameter configurations and feature rankings. Error bars show standard deviations over 30 K-means runs. Dashed KM and dash-dotted GHK-nHSIC-FW lines denote full-feature and continuous-weighting references.

4.3. Cross-Dataset Statistical Comparison

Following cross-dataset comparison procedures [48,49], the eight methods are ranked within each dataset, assigning average ranks to ties. Table 5 reports mean ranks, and Table 6 gives tie-corrected Friedman tests with asymptotic and permutation-based p-values. GHK-nHSIC-FW has average rank 1.5 on both metrics. The omnibus tests indicate differences among methods under the stated benchmark protocol.

Table 5. Mean ranks across the eight datasets (smaller is better).

Method	ACC Rank	NMI Rank
KM	7.125	6.188
LS-W	5.000	4.750
LLSRFS-W	4.000	3.625
MCFS-W	4.875	4.938
UHSIC-W	5.375	5.500
NDFS-W	4.250	5.000
U-HSIC-ENet	3.875	4.500
GHK-nHSIC-FW	1.500	1.500

Table 6. Friedman tests across eight methods and eight datasets.

Metric	Friedman χ^2	Asymptotic p	Permutation p
ACC	23.8084	0.001231	0.000285
NMI	19.4301	0.006942	0.003350

Pairwise comparisons use one-sided Wilcoxon signed-rank tests [50], testing whether GHK-nHSIC-FW has higher dataset-level scores than each baseline. Holm's method [51] corrects seven comparisons within each metric. The paired units are the eight datasets, not the individual K-means runs. Table 7 gives the pairwise results, including win/tie/loss counts.

Table 7. One-sided Wilcoxon comparisons with within-metric Holm adjustment.

Metric	Baseline	Win/Tie/Loss	One-Sided p	Holm p
ACC	KM	8/0/0	0.003906	0.027344
ACC	LS-W	8/0/0	0.003906	0.027344
ACC	LLSRFS-W	8/0/0	0.003906	0.027344
ACC	MCFS-W	7/0/1	0.011719	0.035156
ACC	UHSIC-W	7/0/1	0.011719	0.035156
ACC	NDFS-W	7/0/1	0.019531	0.035156
ACC	U-HSIC-ENet	7/0/1	0.007812	0.031250
NMI	KM	8/0/0	0.003906	0.027344
NMI	LS-W	8/0/0	0.003906	0.027344
NMI	LLSRFS-W	6/0/2	0.039062	0.039062
NMI	MCFS-W	8/0/0	0.003906	0.027344
NMI	UHSIC-W	8/0/0	0.003906	0.027344
NMI	NDFS-W	7/0/1	0.019531	0.039062
NMI	U-HSIC-ENet	7/0/1	0.011719	0.035156

After Holm adjustment, all seven comparisons have adjusted p-values below 0.05 for both ACC and NMI. These results are consistent with the lower mean ranks of the proposed method while retaining the dataset-specific reversals shown in Tables 3 and 4.

4.4. Ablation Study

Three ablations assess the model components. The w/o graph heat kernel variant replaces the diffusion reference with a Gaussian kernel constructed from the full input. The w/o redundancy variant removes the redundancy quadratic term while retaining the relevance term, squared-norm regularization, and simplex constraint. The w/o nHSIC variant uses unnormalized HSIC for both feature–target relevance and feature–feature redundancy. All variants use the same mean ACC parameter selection criterion and downstream K-means evaluation protocol. Table 8 summarizes the macro-average ACC and NMI of the full method and its ablated variants.

Table 8. Mean scores over the eight public datasets for the component ablations.

Method	ACC	NMI
GHK-nHSIC-FW	0.7063	0.4679
w/o Graph Heat Kernel	0.6670	0.4469
w/o Redundancy	0.6794	0.4577
w/o nHSIC	0.6764	0.4505

The full method attains the highest macro-average ACC and NMI among the ablations. Replacing the graph heat kernel target lowers these metrics by 0.0393 and 0.0210. Removing redundancy lowers them by 0.0269 and 0.0102. Replacing nHSIC with unnormalized HSIC lowers them by 0.0299 and 0.0174. These comparisons support the contributions of the target construction, redundancy term, and normalization in the evaluated settings. The controlled feature copy experiment in Section 4.5.3 further examines how redundancy changes the allocation of feature weights.

4.5. Controlled Experiments on Synthetic Data

Four controlled experiments examine sample contamination, irrelevant coordinates, correlated feature copies, and changes to the initial graph. Each condition is evaluated on ten independently generated datasets with ten K-means random seeds per dataset. The first three experiments use the predefined parameter grids and mean ACC selection rule in Section 4.1; the fourth examines reference kernels across diffusion scales. Clustering scores are averaged over K-means seeds within each dataset and then summarized across the ten datasets. Where shown, shaded bands indicate one standard deviation across datasets. Within each replicate, methods share the same data, and perturbation levels share the underlying clean samples and random feature pools. Each resulting dataset is Z-score standardized before feature learning and clustering.

Each clean dataset contains three classes of 100 samples and four coordinates. Class-conditional coordinates are independent Gaussian variables with mean vectors $(0.5,-0.5,0,0)$, $(-0.5,-0.5,0,0)$, and $(0,0.5,0,0)$ and standard deviation vectors $(0.2,0.2,1,1)$, $(0.2,0.2,1,1)$, and $(0.6,0.2,1,1)$, respectively. Features f_1 and f_2 are informative, whereas f_3 and f_4 are independent standard normal nuisance coordinates. The larger variance of f_1 in the third class increases class overlap.

4.5.1. Uniform Background Sample Contamination

Uniform background contamination adds samples drawn independently from $U(-3,3)^4$. For nominal contamination rate $\rho \in \{0,0.1,0.2,0.3,0.4\}$, the number added is $n_o = \text{round}(300\rho/(1-\rho))$, giving 0, 33, 75, 129, and 200 contaminating samples. All samples enter standardization, feature learning, and K-means; ACC is computed only on the 300 original clean samples. Figure 3 reports clean sample ACC and total informative feature weight. GHK-nHSIC-FW has the highest mean ACC at the nonzero contamination levels

and retains a larger informative feature weight share than the displayed alternatives at the highest level. Performance declines as contamination increases. Since standardization also uses the contaminating samples, this experiment measures the effect of contamination on the complete preprocessing and weighting pipeline.

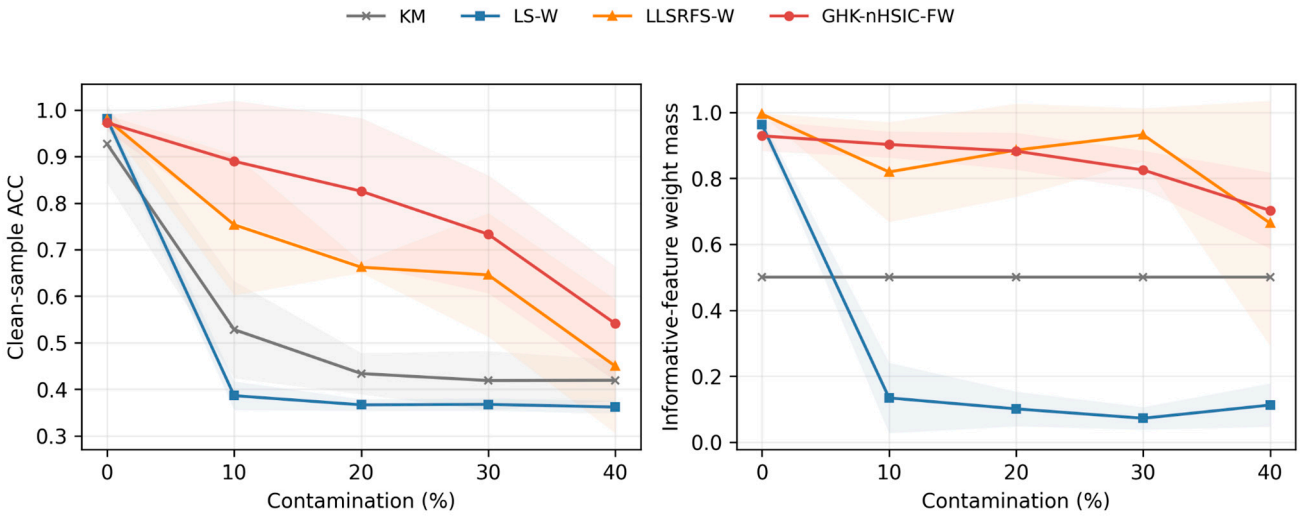


Figure 3. Uniform background contamination: clean sample ACC (**left**) and total weight on the two informative coordinates (**right**). Lines show means and shaded bands show one standard deviation across ten independent datasets.

4.5.2. Irrelevant Feature Injection

We append $q \in \{0,5,10,20,40\}$ independent $N(0,1)$ noise coordinates to the four-dimensional clean data. Within each replicate, the first q columns of a common noise feature pool are used, producing nested perturbations. Figure 4 reports ACC, the total weight on the original informative coordinates, and the fraction of initial KNN edges connecting samples from the same true class. Labels are used to measure this graph diagnostic, not to construct the graph. Increasing irrelevant dimensionality reduces graph class consistency and clustering performance. At 40 added coordinates, GHK-nHSIC-FW retains the highest mean ACC and informative feature weight among the displayed methods, although both decline substantially from the clean condition.

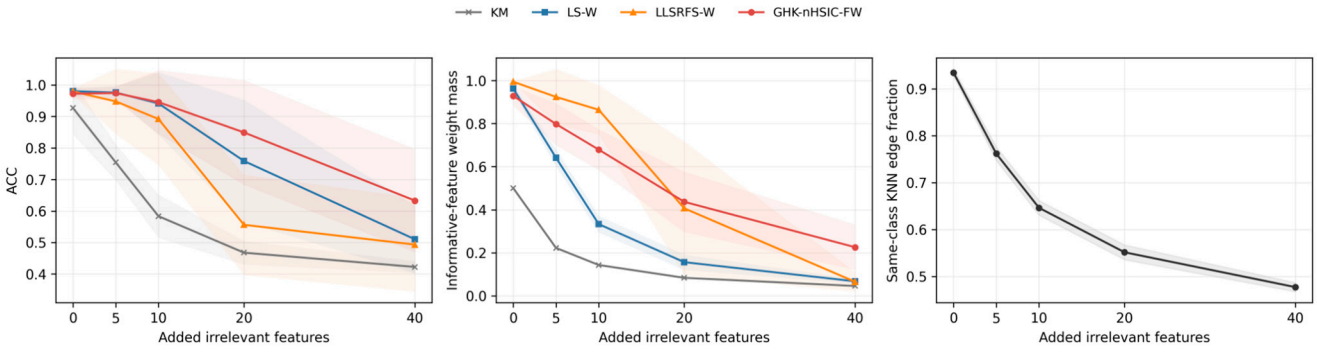


Figure 4. Irrelevant feature injection: ACC (**left**), weight on the two original informative coordinates (**middle**), and same class KNN edge fraction (**right**). Shaded bands show one standard deviation across ten datasets.

4.5.3. Correlated Feature Copies and Redundancy Control

We append $q \in \{0,1,3,5,10\}$ noisy copies of the first informative coordinate, defined with $\tilde{f}_{1,j} = f_1 + 0.1\mathrm{std}(f_1)\epsilon_j$, where $\epsilon_j \sim N(0,1)$ independently. These features are highly

correlated with f_1 but are not exact duplicates. Figure 5 reports ACC, the combined weight of f_1 and its copies, and the separate weight of f_2. The full method and its variant without redundancy have similar ACC with no added copies. Under duplication, the variant places more weight on the repeated group and markedly less on f_2, with lower ACC than the full method. The full model maintains a more balanced allocation between the correlated group and the distinct informative coordinate. This directly supports the role of the redundancy term in the tested feature copy setting.

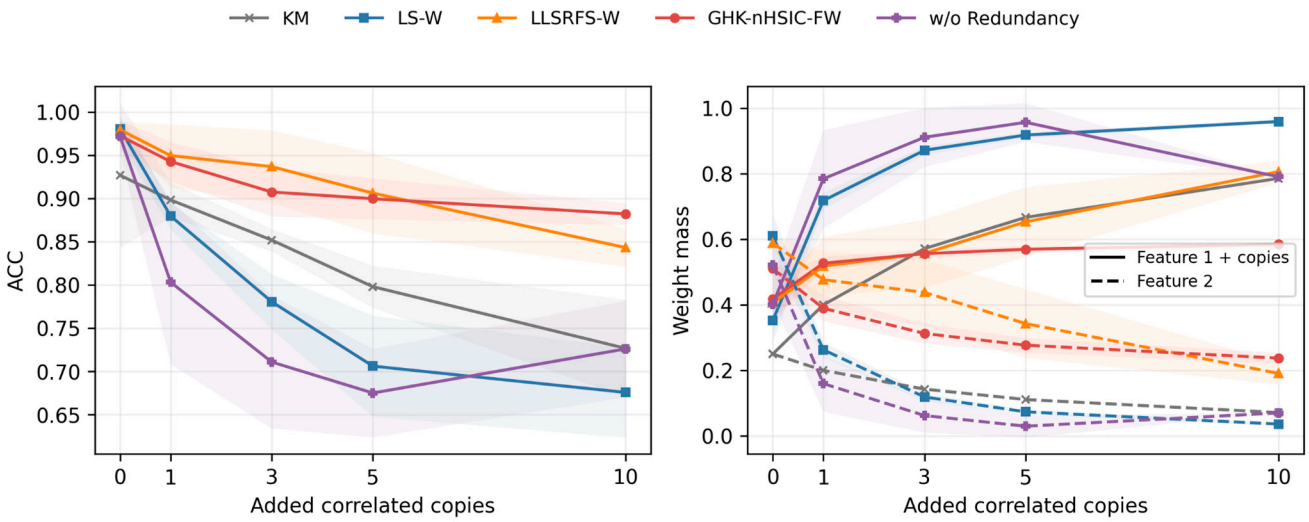


Figure 5. Correlated informative feature copies: ACC (**left**) and feature group weights (**right**). Solid lines give the total weight of the first feature and its copies; dashed lines give the weight of the second informative feature. Shaded bands show one standard deviation across ten datasets.

4.5.4. Effect of Diffusion Scale Under Graph Perturbations

The samples and features are held fixed while $p \in \{0, 0.05, 0.10, 0.20, 0.30\}$ of the undirected non-self edges in the weighted union-10NN graph are randomly replaced by previously absent edges. Nested perturbations preserve the edge count, edge-weight multiset, symmetry, and unit self-loops, while avoiding isolated vertices. With $S = D^{-1/2}WD^{-1/2}$, we compare the local neighborhood-profile kernel $K_{\mathrm{local}} = SS^{\mathrm{T}} = S^2$ with $K_{\mathrm{heat}}(\tau) = \exp(-\tau L_{\mathrm{sym}})$, for $\tau \in \{0.1, 0.3, 0.5, 1, 2, 5, 10, 20, 50\}$. Neighborhood consistency Q is the mean same-class fraction among each sample's ten most similar non-self neighbors. With $b_j(K) = \mathrm{nHSIC}(K_j, K)$ before Min–Max scaling, the relevance gap is $G(K) = [b_1(K) + b_2(K) - b_3(K) - b_4(K)]/2$. Known classes and informative coordinates are used only to compute these diagnostics. ACC is evaluated after applying the same feature-weighting objective to each reference. Figure 6 reports mean paired differences (heat minus local) over ten datasets.

At $\tau \leq 2$, all three mean differences are negative at every perturbation level. At $\tau = 5$ and 10, all three are positive; Q and G improve on all ten datasets at every perturbation level, whereas ACC improves on seven to ten datasets. Larger scales show a different pattern. At $\tau = 20$, ΔQ becomes negative under 20% and 30% edge replacement, while ΔG and ΔACC remain positive. At $\tau = 50$, neighborhood consistency deteriorates under all nonzero perturbation levels, although the other metrics do not decline uniformly. Thus, moderate diffusion improves both structural diagnostics and downstream clustering in this setting, but clustering gains alone do not establish more accurate reference neighborhoods. ACC also reflects the subsequent relevance scaling and weight optimization.

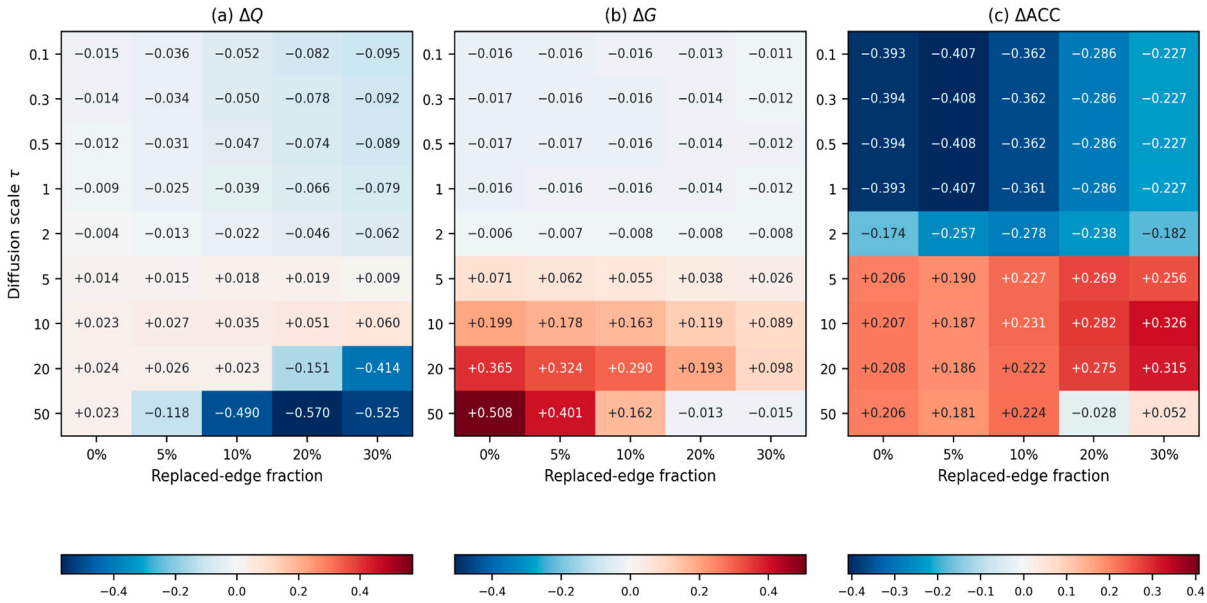


Figure 6. Paired differences between heat and local neighborhood profile references: (**a**) neighborhood consistency ΔQ, (**b**) raw nHSIC relevance gap ΔG, and (**c**) downstream ΔACC. Rows indicate diffusion scales and columns indicate replaced-edge fractions. Each cell gives the mean difference over ten independent datasets, rounded to three decimal places. Positive values favor the heat kernel. Color scales are centered at zero and shown separately for each panel.

4.6. Subsampling Stability of Feature Importance

For each of the eight public datasets, 30 subsets are sampled without replacement at 80% of the original sample size, using shared subset indices across methods. Feature importance is relearned with the already selected full-data hyperparameters and without clustering-based retuning, preserving the original feature standardization scale. Each subset importance vector is compared with the full-data reference using Spearman rank correlation, cosine similarity, and Jaccard overlap of the top 20% ranked features [52]. For Jaccard@20%, ties at the selection boundary are retained; consequently, selected sets may exceed the nominal budget. All methods complete the 240 requested subset comparisons. Macro-averages are obtained by first averaging within each dataset and then giving equal weight to the eight dataset means.

The proposed method reaches mean Spearman correlation 0.9016 and cosine similarity 0.9848, both second to LS, and Jaccard@20% of 0.7888. These results show high rank and vector direction consistency over the subsampling experiment. Figure 7 displays the distributions of the eight dataset-level mean stability scores, complementing the macro-averages in Table 9.

Table 9. Macro-average feature importance stability across eight datasets.

Method	Spearman	Cosine	Jaccard@20%
LS	**0.9262**	**0.9906**	**0.8430**
LLSRFS	0.7835	0.8483	0.7781
MCFS	0.5518	0.7139	0.4547
UHSIC-Lasso	0.8551	0.9192	0.7569
NDFS	0.7040	0.7797	0.5927
U-HSIC-ENet	0.7694	0.8357	0.8275
GHK-nHSIC-FW	0.9016	0.9848	0.7888

Note: Bold indicates the highest mean in each column.

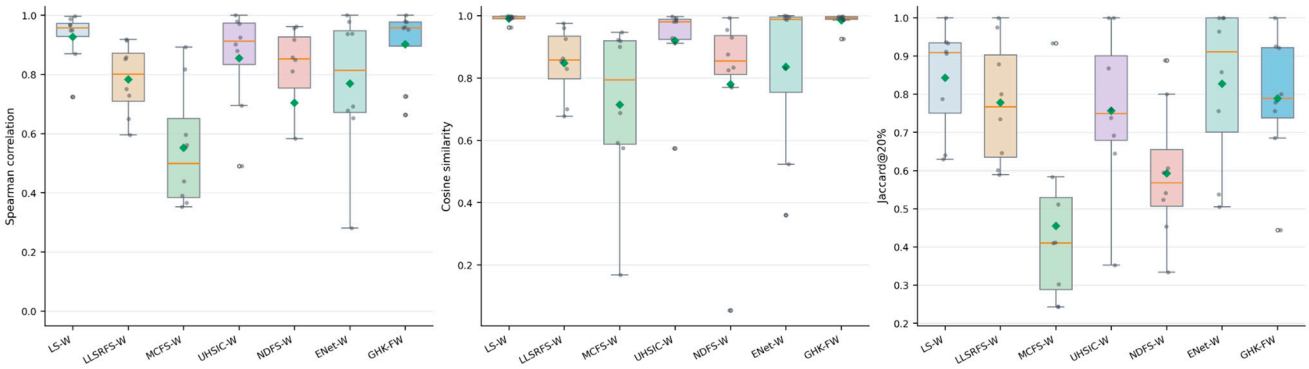


Figure 7. Subsampling stability across eight public datasets for Spearman correlation, cosine similarity, and Jaccard@20%. Each box represents eight dataset-level mean scores. Gray dots denote individual datasets, orange lines indicate medians, and green diamonds indicate means.

5. Conclusions and Future Work

We introduced GHK-nHSIC-FW, a filter-based unsupervised feature-weighting method that combines a graph heat kernel structural target, nHSIC relevance and redundancy measures, and simplex-constrained quadratic optimization. The learned continuous weights define a distance representation for downstream clustering without using cluster assignments during weight learning at fixed hyperparameters.

On eight public datasets, GHK-nHSIC-FW achieves the best macro-average ACC (0.7063) and NMI (0.4679) under a unified continuous-weighted K-means protocol. Cross-dataset statistical tests and component ablations support the framework. Controlled experiments show the behavior of informative feature weights and clustering under background sample contamination, irrelevant feature injection, correlated feature duplication, and diffusion-scale effects under neighborhood graph perturbation. Subsampling results indicate high feature importance stability, including Spearman correlation 0.9016 and cosine similarity 0.9848.

The usefulness of the diffusion target depends on the initial neighborhood graph, and exact kernel construction limits scalability. Future work will investigate adaptive graph construction, computationally efficient kernel approximations, and practical label-free strategies for selecting model hyperparameters.

Author Contributions: Conceptualization, L.Z.; methodology, L.Z. and T.W.; software, L.Z.; validation, L.Z., Y.C., and T.W.; resources, Y.C.; data curation, L.Z. and Y.C.; writing—original draft preparation, L.Z.; writing—review and editing, L.Z., Y.C., and T.W.; supervision, T.W.; project administration, T.W.; funding acquisition, Y.C. and T.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported in part by the National Natural Science Foundation of China (No. 62666004), Natural Science Foundation of Jiangxi Province of China (No. 20242BAB26024), and the Graduate Innovation Special Foundation of Jiangxi Province of China (No. YC2025-S759).

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon request. There are no restrictions on data availability.

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

References

1. Guyon, I.; Elisseeff, A. An introduction to variable and feature selection. *J. Mach. Learn. Res.* **2003**, *3*, 1157–1182. [CrossRef]
2. Solório-Fernández, S.; Carrasco-Ochoa, J.A.; Martínez-Trinidad, J.F. A review of unsupervised feature selection methods. *Artif. Intell. Rev.* **2020**, *53*, 907–948. [CrossRef]
3. Hancer, E.; Xue, B.; Zhang, M. A survey on feature selection approaches for clustering. *Artif. Intell. Rev.* **2020**, *53*, 4519–4545. [CrossRef]
4. de Amorim, R.C. A survey on feature weighting based K-means algorithms. *J. Classif.* **2016**, *33*, 210–242. [CrossRef]
5. Niño-Adan, I.; Manjarres, D.; Landa-Torres, I.; Portillo, E. Feature weighting methods: A review. *Expert Syst. Appl.* **2021**, *184*, 115424. [CrossRef]
6. Huang, J.Z.; Ng, M.K.; Rong, H.; Li, Z. Automated variable weighting in k-means type clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 657–668. [CrossRef] [PubMed]
7. Du, X. A robust and high-dimensional clustering algorithm based on feature weight and entropy. *Entropy* **2023**, *25*, 510. [CrossRef] [PubMed]
8. Xiong, L.; Wang, C.; Huang, X.; Zeng, H. An entropy regularization k-means algorithm with a new measure of between-cluster distance in subspace clustering. *Entropy* **2019**, *21*, 683. [CrossRef] [PubMed]
9. Modha, D.S.; Spangler, W.S. Feature weighting in k-means clustering. *Mach. Learn.* **2003**, *52*, 217–237. [CrossRef]
10. Jing, L.; Ng, M.K.; Huang, J.Z. An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data. *IEEE Trans. Knowl. Data Eng.* **2007**, *19*, 1026–1041. [CrossRef]
11. He, X.; Cai, D.; Niyogi, P. Laplacian score for feature selection. In *Advances in Neural Information Processing Systems 18*; MIT Press: Cambridge, MA, USA, 2006; pp. 507–514.
12. Zhao, Z.; Liu, H. Spectral feature selection for supervised and unsupervised learning. In Proceedings of the 24th International Conference on Machine Learning (ICML 2007), Corvallis, OR, USA, 20–24 June 2007; pp. 1151–1157. [CrossRef]
13. Cai, D.; Zhang, C.; He, X. Unsupervised feature selection for multi-cluster data. In Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 25–28 July 2010; pp. 333–342. [CrossRef]
14. Qian, M.; Zhai, C. Robust unsupervised feature selection. In Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, 3–9 August 2013; pp. 1621–1627.
15. Chen, H.; Nie, F.; Wang, R.; Li, X. Unsupervised feature selection with flexible optimal graph. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *35*, 2014–2027. [CrossRef] [PubMed]
16. Zuo, X.; Zhang, W.; Wang, X.; Dang, L.; Qiao, B.; Wang, Y. Unsupervised feature selection via maximum relevance and minimum global redundancy. *Pattern Recognit.* **2025**, *164*, 111483. [CrossRef]
17. Nie, F.; Zhu, W.; Li, X. Unsupervised feature selection with structured graph optimization. *Proc. AAAI Conf. Artif. Intell.* **2016**, *30*, 1302–1308. [CrossRef]
18. Wang, C.; Wang, J.; Gu, Z.; Wei, J.-M.; Liu, J. Unsupervised feature selection by learning exponential weights. *Pattern Recognit.* **2024**, *148*, 110183. [CrossRef]
19. Du, L.; Shen, Y.-D. Unsupervised feature selection with adaptive structure learning. In Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, Australia, 10–13 August 2015; pp. 209–218. [CrossRef]
20. Banerjee, M.; Pal, N.R. Unsupervised feature selection with controlled redundancy (UFeSCoR). *IEEE Trans. Knowl. Data Eng.* **2015**, *27*, 3390–3403. [CrossRef]
21. Mitra, P.; Murthy, C.A.; Pal, S.K. Unsupervised feature selection using feature similarity. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 301–312. [CrossRef]
22. Yan, Z.; Ma, Z.; Ma, J.; Li, H. Robust unsupervised feature selection algorithm based on fuzzy anchor graph. *Entropy* **2025**, *27*, 827. [CrossRef] [PubMed]
23. Gretton, A.; Bousquet, O.; Smola, A.; Schölkopf, B. Measuring statistical dependence with Hilbert-Schmidt norms. In Proceedings of the 16th International Conference on Algorithmic Learning Theory, Singapore, 8–11 October 2005; pp. 63–77. [CrossRef]
24. Yamada, M.; Jitkrittum, W.; Sigal, L.; Xing, E.P.; Sugiyama, M. High-dimensional feature selection by feature-wise kernelized Lasso. *Neural Comput.* **2014**, *26*, 185–207. [CrossRef] [PubMed]
25. Wang, T. A survey on Hilbert-Schmidt independence criterion Lasso. *Knowl.-Based Syst.* **2026**, *342*, 115934. [CrossRef]
26. Wang, T.; Hu, Z.; Liu, H. A unified view of feature selection based on Hilbert-Schmidt independence criterion. *Chemom. Intell. Lab. Syst.* **2023**, *236*, 104807. [CrossRef]
27. von Luxburg, U. A tutorial on spectral clustering. *Stat. Comput.* **2007**, *17*, 395–416. [CrossRef]
28. Smola, A.J.; Kondor, R. Kernels and regularization on graphs. In Proceedings of the 16th Annual Conference on Computational Learning Theory and 7th Kernel Workshop, Washington, DC, USA, 24–27 August 2003; pp. 144–158. [CrossRef]

29. Belkin, M.; Niyogi, P. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.* **2003**, *15*, 1373–1396. [CrossRef]
30. Kondor, R.I.; Lafferty, J. Diffusion kernels on graphs and other discrete structures. In Proceedings of the 19th International Conference on Machine Learning, Sydney, Australia, 8–12 July 2002; pp. 315–322.
31. Coifman, R.R.; Lafon, S. Diffusion maps. *Appl. Comput. Harmon. Anal.* **2006**, *21*, 5–30. [CrossRef]
32. Hammond, D.K.; Vandergheynst, P.; Gribonval, R. Wavelets on graphs via spectral graph theory. *Appl. Comput. Harmon. Anal.* **2011**, *30*, 129–150. [CrossRef]
33. Guo, J.; Zhu, W. Dependence guided unsupervised feature selection. *Proc. AAAI Conf. Artif. Intell.* **2018**, *32*, 2232–2239. [CrossRef]
34. Li, Z.; Yang, Y.; Liu, J.; Zhou, X.; Lu, H. Unsupervised feature selection using nonnegative spectral analysis. *Proc. AAAI Conf. Artif. Intell.* **2012**, *26*, 1026–1032. [CrossRef]
35. Jain, A.K. Data clustering: 50 years beyond K-means. *Pattern Recognit. Lett.* **2010**, *31*, 651–666. [CrossRef]
36. Epstein, E.; Nallapareddy, N.; Ray, S. On the relationship between feature selection metrics and accuracy. *Entropy* **2023**, *25*, 1646. [CrossRef] [PubMed]
37. Wang, T.; Dai, X.; Liu, Y. Learning with Hilbert-Schmidt independence criterion: A review and new perspectives. *Knowl.-Based Syst.* **2021**, *234*, 107567. [CrossRef]
38. Vergara, J.R.; Estévez, P.A. A review of feature selection methods based on mutual information. *Neural Comput. Appl.* **2014**, *24*, 175–186. [CrossRef]
39. Sugiyama, M. Machine learning with squared-loss mutual information. *Entropy* **2013**, *15*, 80–112. [CrossRef]
40. Song, L.; Bedo, J.; Borgwardt, K.M.; Gretton, A.; Smola, A. Gene selection via the BAHSIC family of algorithms. *Bioinformatics* **2007**, *23*, i490–i498. [CrossRef] [PubMed]
41. Song, L.; Smola, A.; Gretton, A.; Bedo, J.; Borgwardt, K. Feature selection via dependence maximization. *J. Mach. Learn. Res.* **2012**, *13*, 1393–1434.
42. Cortes, C.; Mohri, M.; Rostamizadeh, A. Algorithms for learning kernels based on centered alignment. *J. Mach. Learn. Res.* **2012**, *13*, 795–828.
43. Liaghat, S.; Mansoori, E.G. Filter-based unsupervised feature selection using Hilbert-Schmidt independence criterion. *Int. J. Mach. Learn. Cybern.* **2019**, *10*, 2313–2328. [CrossRef]
44. Chen, Y.; Wang, T.; Zou, L. Unsupervised feature selection via self-supervised HSIC and elastic net regularization. *Entropy* **2026**, *28*, 1027. [CrossRef] [PubMed]
45. Masoomi, A.; Wu, C.; Zhao, T.; Wang, Z.; Castaldi, P.; Dy, J. Instance-wise feature grouping. In *Advances in Neural Information Processing Systems*; Curran Associates: Red Hook, NY, USA, 2020; Volume 33. Available online: https://proceedings.neurips.cc/paper/2020/hash/9b10a919ddeb07e103dc05ff523afe38-Abstract.html (accessed on 8 November 2026).
46. Kelly, M.; Longjohn, R.; Nottingham, K. The UCI Machine Learning Repository. Available online: https://archive.ics.uci.edu (accessed on 10 October 2026).
47. Strehl, A.; Ghosh, J. Cluster ensembles—A knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.* **2002**, *3*, 583–617.
48. Demšar, J. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* **2006**, *7*, 1–30.
49. Friedman, M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *J. Am. Stat. Assoc.* **1937**, *32*, 675–701. [CrossRef]
50. Wilcoxon, F. Individual comparisons by ranking methods. *Biom. Bull.* **1945**, *1*, 80–83. [CrossRef]
51. Holm, S. A simple sequentially rejective multiple test procedure. *Scand. J. Stat.* **1979**, *6*, 65–70.
52. Nogueira, S.; Sechidis, K.; Brown, G. On the stability of feature selection algorithms. *J. Mach. Learn. Res.* **2018**, *18*, 1–54. [CrossRef]

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