mathematics

MDPI

Article

Multi-View Clustering via Projection-Enhanced Bipartite Graph Learning and Consensus Fusion

Xun Liu [1], Qing-Wen Wang [1,2,*] and Jiang-Feng Chen [1]

[1] Department of Mathematics and Newtouch Center for Mathematics, Shanghai University, Shanghai 200444, China; liuxun@shu.edu.cn (X.L.); cjf23820009@shu.edu.cn (J.-F.C.)
[2] Collaborative Innovation Center for the Marine Artificial Intelligence, Shanghai 200444, China
* Correspondence: wqw@shu.edu.cn

Abstract

Anchor-based bipartite graph methods provide scalable solutions for multi-view clustering, but most of them construct graphs in the original feature space, where high dimensionality distorts the proximity between samples and anchors and degrades graph quality. In addition, the K-means step commonly used to discretize spectral embeddings may produce different cluster assignments across random seeds. To address these limitations, this paper proposes projection-enhanced bipartite graph learning (PEBGL), which first projects each view onto a compact PCA subspace and then jointly performs bipartite graph construction, consensus graph fusion with adaptive view weighting, spectral embedding, and discrete label assignment within an alternating optimization framework. Most subproblems admit closed-form or efficient projection-based updates, and the final labels are obtained by connected-component detection on the learned consensus graph, reducing the dependence on K-means post-processing. Experiments on six benchmark datasets demonstrate that PEBGL achieves competitive clustering performance against recent graph-based and bipartite graph-based methods. These results validate the effectiveness of the proposed framework.

Keywords: multi-view clustering; bipartite graph learning; consensus graph fusion; anchor-based method; spectral embedding

MSC: 62H30

1. Introduction

Driven by advances in artificial intelligence and data-driven optimization, heterogeneous multi-source data have become increasingly common in real-world applications [1,2]. In such scenarios, the same set of objects can often be described by several different types of features. For example, an image can be represented by its color, texture, and shape at the same time. A web document can be described by both its text content and its link structure. A clinical patient may have genomic data, pathological images, and electronic health records. Each type of feature is called a *view*, and each view captures only part of the information about the data. No single view is likely to reveal the complete cluster structure on its own. Multi-view clustering aims to combine these different views to find meaningful groups of objects without any label information. This task has wide applications in areas such as social network analysis, medical informatics, remote sensing, and visual object categorization [3,4].

Among the various approaches to this task, graph-based methods have attracted considerable attention. The basic idea is to build a graph for each view, where nodes represent

Check for updates

Academic Editor: Janez Žerovnik

Received: 17 April 2026
Revised: 10 May 2026
Accepted: 19 May 2026
Published: 21 May 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.

data samples and edge weights reflect pairwise similarities. The cluster structure is then extracted from the spectral properties of the graph. In classical spectral clustering [5], the eigenvectors of the graph Laplacian provide a continuous approximation of the discrete cluster indicators, and K-means is applied as a post-processing step to produce hard assignments. Multi-view extensions build one graph per view and try to learn a unified representation that combines information across views. Nie et al. [6] proposed a graph learning framework that assigns view weights automatically through self-optimization. Wang et al. [7] introduced a mutual reinforcement mechanism between individual view graphs and the consensus graph. Liang et al. [8] modeled both consistency and inconsistency across views, and Huang et al. [9] separated multi-view graphs into shared and diverse components. These methods have significantly improved clustering quality, but they all rely on full $N \times N$ similarity matrices. This requires at least $\mathcal{O}(N^2)$ storage and $\mathcal{O}(N^3)$ eigendecomposition cost, which becomes impractical when N exceeds a few thousand.

Dense pairwise graphs carry a heavy computational burden. To resolve this, researchers select a small set of representative anchors, denoted as r ($r \ll N$). By learning an $r \times N$ affinity matrix between the samples and these anchors, a sparse bipartite structure is formed. Graph construction and spectral analysis both scale as $\mathcal{O}(Nr)$. Li et al. [10] initially adapted this concept for multi-view data. They constructed a separate graph per view and concatenated them prior to spectral analysis. A related pipeline by Kang et al. [11] accelerates subspace clustering. Addressing a different issue, Nie et al. [12] designed a structured objective based on graph connectivity. This yields cluster labels directly, avoiding the K-means step altogether. Meanwhile, other studies [13–15] proposed fusion strategies without additional hyperparameters. These methods aggregate the individual bipartite graphs under a strict Laplacian rank constraint, which guarantees the consensus graph forms exactly c connected components.

A recent trend attempted to solve graph construction and clustering in a single step. Fang et al. [16] reconstructed subspaces to learn bipartite graphs for each view alongside a consensus graph. They also added label learning directly into this loop. Liu et al. [17] relied on decomposition. By splitting each anchor graph into a shared core and a separate residual, they penalized the residuals to reduce discrepancies across views. Yan et al. [18] pushed this idea further. They isolated consistency from diversity, applying sparsity rules so that only the consistent components shaped the final graph. However, these methods all build graphs in the original feature space without modifying the data representation beforehand. Li et al. [19] took a different route by fusing similarities during the spectral embedding phase. By utilizing entropy weighting and spectral rotation, their model produces discrete labels without extra clustering steps. This orthogonal rotation mechanism to replace K-means was originally designed by Luo et al. [20] for multigraph clustering. Several recent bipartite graph models have since adopted this idea [16,19].

Tensors provide a parallel approach to capture high-order correlations across views. Xia et al. [21] stacked view-specific bipartite graphs into a third-order tensor and imposed the tensor Schatten p-norm to encourage low-rank consistency. Gu et al. [22] avoided explicit tensor decomposition by learning a compact essential representation. More recently, Long et al. [23] developed a scalable multi-view tensor clustering method, and Liu et al. [24] further studied large-scale multi-view tensor clustering with implicit linear kernels. Zhao et al. [25] captured indirect relationships between samples and introduced a truncation mechanism to filter out low-quality graphs before fusion. Jiang et al. [26] extended the bipartite graph framework to the unaligned setting, where sample correspondences across views are unknown. Ji et al. [27] further combined tensorized modeling with multi-scale representation learning for unaligned multi-view clustering.

Despite this progress, two limitations remain in existing methods. The first limitation concerns the feature space in which bipartite graphs are constructed. Virtually all methods cited above learn anchor-sample affinities in the original feature space. When a view contains thousands of features, many of which are redundant or uninformative, pairwise distances suffer from the concentration phenomenon in high-dimensional spaces and become nearly indistinguishable. The resulting affinity values lose discriminative power, and the distorted affinities propagate through consensus fusion into the final partition. Although dimensionality reduction techniques such as PCA have long been available, integrating them into the bipartite graph learning objective while preserving the variance structure and maintaining closed-form updates remains unexplored.

The second limitation lies in the transition from continuous spectral embeddings to discrete cluster assignments. Three competing strategies coexist. The dominant approach applies K-means to the eigenvectors of the consensus Laplacian, which introduces nondeterminism through random seed selection and can yield different partitions across runs on the same graph. The constrained Laplacian rank approach [28] enforces exactly c connected components, yielding deterministic labels from the connectivity pattern, but it imposes a rigid structural requirement that may not suit data whose clusters overlap or vary in density. Spectral rotation [29] aligns the eigenvector matrix to a discrete indicator through orthogonal rotation, removing K-means entirely, yet it does not by itself ensure that the underlying graph possesses a clear connected-component structure. Each strategy sacrifices one desirable property to gain another, and combining their strengths within a unified objective remains an open question.

We propose projection-enhanced bipartite graph learning (PEBGL) to address these limitations. The method first projects each view onto a compact PCA subspace and then jointly performs bipartite graph construction, entropy-regularized consensus fusion, spectral embedding, and discrete rotation within an alternating optimization framework. In the projected space, anchor reconstruction under simplex constraints forms the bipartite graph. By applying an entropy penalty, the method learns adaptive weights for the view-specific graphs and fuses them into one consensus structure. A spectral embedding is computed from the symmetric normalized Laplacian of the consensus graph, and an orthogonal rotation aligns this embedding with discrete cluster indicators as a structural regularizer. Final labels are obtained from the connected components of the converged consensus graph, reducing the dependence on K-means post-processing. Experiments on six benchmark datasets demonstrate the competitive performance and computational efficiency of PEBGL.

(i) A projection-enhanced bipartite graph learning framework is proposed, which combines fixed PCA-based projection preprocessing with scalable sample-anchor graph learning.
(ii) An entropy-regularized consensus fusion strategy is developed to adaptively integrate view-specific bipartite graphs into a unified consensus graph.
(iii) A K-means-free final decoding scheme is introduced by detecting connected components on the learned consensus graph, reducing the dependence on K-means post-processing.

The overall framework of PEBGL is depicted in Figure 1.

Table 1 summarizes the key symbols used throughout this paper. The set of $N \times c$ binary matrices in which each row contains exactly one unit entry is denoted by Ind. Let $\{X^v\}_{v=1}^{V}$ denote a multi-view dataset with V views, where $X^v \in \mathbb{R}^{N \times d_v}$ contains N samples described by d_v-dimensional features in the v-th view. The goal is to partition the N samples into c clusters by jointly exploiting all views.

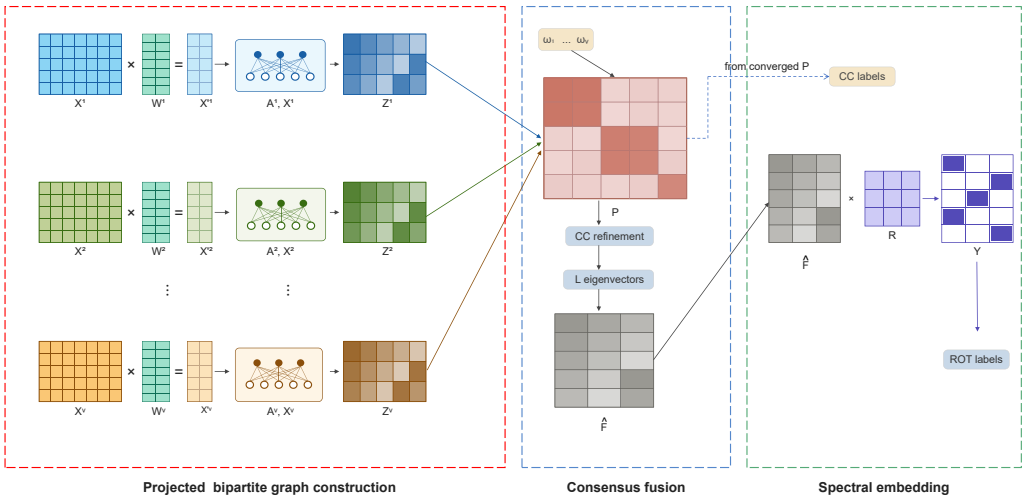


Figure 1. Overview of the PEBGL framework. Each view X^v is projected into a compact subspace via fixed PCA projection W^v, within which view-specific bipartite graphs Z^v are constructed through anchor-based reconstruction. The view-specific graphs are fused into a consensus graph P through entropy-regularized adaptive weighting ω_v, followed by coclustering refinement. A spectral embedding $\hat{F}$ is extracted from the normalized Laplacian of P and aligned to a discrete indicator Y through orthogonal rotation R, which serves as a structural regularizer during optimization. The final cluster labels are read from the connected components of the converged P.

Table 1. Summary of key notations.

Symbol	Description
N, V, c, r	Number of samples, views, clusters, and anchors
d_v, d'_v	Original and projected feature dimension of view v
X^v	Feature matrix of view v, $X^v \in \mathbb{R}^{N \times d_v}$
A^v	Anchor matrix of view v, $A^v \in \mathbb{R}^{r \times d_v}$
W^v	PCA projection matrix, $(W^v)^\top W^v = I_{d'_v}$
$\tilde{X}^v, \tilde{A}^v$	Projected data and anchor matrices
Z^v	Bipartite graph of view v, $Z^v \in \mathbb{R}^{r \times N}$
P	Consensus bipartite graph, $P \in \mathbb{R}^{r \times N}$
$\hat{F}$	Spectral embedding matrix, $\hat{F} \in \mathbb{R}^{N \times c}$
R	Orthogonal rotation matrix, $R^\top R = I_c$
Y	Discrete indicator matrix, $Y \in$ Ind
ω_v	Adaptive weight for view v
L_P	Symmetric normalized Laplacian derived from P

The remainder of this paper is organized as follows. Section 2 establishes the mathematical preliminaries. Section 3 develops the proposed method with complete derivations. Section 4 reports experimental comparisons and analyses on six benchmark datasets. Section 5 concludes the paper.

2. Preliminaries

This section reviews the mathematical tools that the proposed method builds upon.

2.1. Graph Laplacian

Let W be a symmetric non-negative similarity matrix with degree matrix $D = \mathrm{diag}(W\mathbf{1})$. The symmetric normalized Laplacian is

$$L = I - D^{-1/2} W D^{-1/2}. \tag{1}$$

Lemma 1 (von Luxburg [30]). *The multiplicity of the eigenvalue* 0 *of* L *equals the number of connected components in the graph associated with* W.

When W encodes a bipartite graph between N samples and r anchors through a non-negative matrix Z, the augmented similarity matrix takes the block form

$$S = \begin{bmatrix} 0 & Z^\top \\ Z & 0 \end{bmatrix}, \tag{2}$$

and Lemma 1 applies to its normalized Laplacian $\tilde{L}_S = I - D_S^{-1/2} S D_S^{-1/2}$, where D_S is the diagonal degree matrix. If $\tilde{L}_S$ has exactly c zero eigenvalues, then S possesses exactly c connected components, each defining a cluster. This observation motivates the coclustering decoding adopted in Section 3.

In the standard spectral clustering pipeline [5], the c eigenvectors of L associated with the smallest eigenvalues are collected into F with $F^\top F = I_c$, and K-means is applied to the rows of F to produce cluster labels. A multi-view extension replaces the single Laplacian with a weighted combination $\sum_{v=1}^{V} \omega_v L^v$, and the embedding is obtained by minimizing

$$\min_{F^\top F = I_c} \sum_{v=1}^{V} \omega_v \operatorname{Tr}\left(F^\top L^v F\right). \tag{3}$$

This formulation is invariant with relabeling of the view indices, provided the weights are permuted accordingly. Assigning adaptive weights and reducing the dependence on K-means post-processing are two design choices that the proposed method addresses.

2.2. Bipartite Graph

Full graph spectral clustering requires $\mathcal{O}(N^2)$ storage and $\mathcal{O}(N^3)$ eigendecomposition time. Bipartite graph methods avoid this bottleneck by introducing r representative anchor points with $r \ll N$ and learning an $r \times N$ affinity matrix instead of an $N \times N$ similarity matrix.

Given a data matrix X and an anchor matrix A, the bipartite graph Z is obtained by expressing each sample as a non-negative combination of the anchors [13,16,17],

$$\min_{Z \geq 0,\, Z^\top \mathbf{1} = \mathbf{1}} \left\| X^\top - A^\top Z \right\|_F^2 + \alpha \|Z\|_F^2, \tag{4}$$

where $\alpha > 0$ is a regularization parameter. The constraint $Z^\top \mathbf{1} = \mathbf{1}$ requires each column of Z to lie on the probability simplex

$$\Delta^{r-1} = \left\{ z \in \mathbb{R}^r \mid z \geq 0,\ \mathbf{1}^\top z = 1 \right\}. \tag{5}$$

The Euclidean projection of an arbitrary vector $y \in \mathbb{R}^r$ onto Δ^{r-1} admits the closed-form solution $z_i^* = \max(y_i - \theta, 0)$, where the threshold θ is uniquely determined by $\sum_i \max(y_i - \theta, 0) = 1$ and can be computed in $\mathcal{O}(r \log r)$ time [31]. Graph construction, storage, and all subsequent operations on Z scale as $\mathcal{O}(Nr)$.

In the multi-view setting, each view v produces a bipartite graph Z^v, and fusing $\{Z^v\}_{v=1}^V$ into a consensus bipartite graph that captures the shared cluster structure is the central algorithmic challenge.

The PCA projection matrix W^v satisfies $(W^v)^\top W^v = I_{d_v'}$, and the rotation matrix R satisfies $R^\top R = I_c$. In the proposed method, W^v is computed once by PCA before graph learning, while R is updated through singular value decomposition, as detailed in Section 3.

2.3. Matrix Inequalities

The optimization steps for the spectral embedding and the rotation matrix use the following standard matrix inequalities.

Lemma 2 (Courant–Fischer min-max theorem [32]). *Let $M \in \mathbb{R}^{n\times n}$ be a symmetric matrix with eigenvalues $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$. For any W with $W^\top W = I_p$ and $p \le n$,*

$$\mathrm{Tr}(W^\top MW) \ge \sum_{k=1}^{p} \lambda_k, \tag{6}$$

and equality holds when the columns of W are the eigenvectors of M corresponding to $\lambda_1, \ldots, \lambda_p$.

This result explains why the eigenvectors associated with the smallest eigenvalues of a graph Laplacian characterize the relaxed spectral embedding used in spectral clustering.

Lemma 3 (von Neumann trace inequality [33]). *Let $A, B \in \mathbb{R}^{m\times n}$ with singular values $\sigma_1(A) \ge \cdots \ge \sigma_{\min(m,n)}(A)$ and $\sigma_1(B) \ge \cdots \ge \sigma_{\min(m,n)}(B)$, respectively. Then*

$$\mathrm{Tr}(A^\top B) \le \sum_{i=1}^{\min(m,n)} \sigma_i(A)\,\sigma_i(B), \tag{7}$$

with equality when $A = U_B V_B^\top$, where $B = U_B \Sigma_B V_B^\top$ is a singular value decomposition of B.

2.4. Spectral Rotation

The eigenvector matrix F obtained from a graph Laplacian is determined only up to multiplication by an arbitrary orthogonal matrix. Two bases F and FQ with $Q^\top Q = I_c$ span the same subspace and encode the same partition information, yet K-means applied to F and to FQ may yield different cluster assignments.

Spectral rotation [29] eliminates this ambiguity. Given F with $F^\top F = I_c$ and a discrete indicator matrix $Y \in \mathrm{Ind}$, the joint optimization

$$\max_{R^\top R = I_c,\, Y \in \mathrm{Ind}} \mathrm{Tr}\Big((FR)^\top Y\Big) \tag{8}$$

aligns the rotated embedding FR with Y by alternating two steps. With Y fixed, the optimal R is obtained by solving the orthogonal Procrustes problem via Lemma 3. With R fixed, the optimal Y is obtained by selecting the largest entry in each row of FR, a deterministic operation that requires no random initialization. The alternating procedure converges monotonically because each step increases the trace objective over its respective feasible set. This mechanism produces identical cluster labels across independent runs for any fixed input F, resolving the reproducibility issue associated with K-means post-processing.

3. Proposed Method

This section presents the PEBGL framework, develops each model component, and derives the alternating optimization procedure.

3.1. Projected Bipartite Graph Learning

Existing anchor-based methods construct bipartite graphs directly in the original feature space. When the feature dimensionality is high, pairwise distances between samples and anchors tend to concentrate, degrading the quality of the resulting affinity graphs. To mitigate this effect, a fixed PCA-based projection preprocessing step is introduced for each view, mapping the original features into a compact subspace before graph construction.

For each view v, a matrix $W^v \in \mathbb{R}^{d_v \times d'_v}$ with orthonormal columns, satisfying $(W^v)^\top W^v = I_{d'_v}$, projects both the data matrix X^v and the anchor matrix A^v into a common low-dimensional space,

$$\tilde{X}^v = X^v W^v, \qquad \tilde{A}^v = A^v W^v. \tag{9}$$

The projected dimension d'_v is determined individually for each view by retaining the leading principal components that account for at least 95% of the cumulative variance. A lower bound of c is imposed to preserve sufficient discriminative capacity. For high-dimensional small-sample datasets with $d_v \gg N$, the projected dimension is capped at c to avoid an underdetermined linear system.

The projection matrix W^v is computed once by PCA before the main alternating optimization and then kept fixed. Thus, the projection is treated as a preprocessing step rather than as a jointly optimized variable in the main loop. The fixed PCA projection maps each view onto a compact subspace that preserves the dominant variance structure, thereby reducing redundant high-dimensional information before sample-anchor affinity construction while keeping the subsequent optimization efficient and numerically stable.

A prerequisite for bipartite graph methods is the construction of semantically aligned anchors across views. Independent per-view anchor generation risks semantic misalignment, where the k-th anchor in one view corresponds to a different data region than the k-th anchor in another view. To ensure consistent anchor semantics, all views are concatenated into $X_{\text{cat}} = [X^1, X^2, \ldots, X^V] \in \mathbb{R}^{N \times D}$ with $D = \sum_v d_v$, and K-means is performed on X_{cat} to obtain r global centroids. These centroids are then split along the feature dimension to recover view-specific anchor matrices A^v, guaranteeing that corresponding anchors share the same global origin across all views.

Given the projected data $\tilde{X}^v$ and projected anchors $\tilde{A}^v$, a view-specific bipartite graph Z^v is constructed to encode the affinity between each sample and the r anchors. Following the anchor-based subspace learning paradigm [16,17], each projected sample is modeled as a non-negative linear combination of the projected anchors, leading to the reconstruction objective

$$\min_{Z^v} \left\| \tilde{X}^{v\top} - \tilde{A}^{v\top} Z^v \right\|_F^2 + \alpha \|Z^v\|_F^2, \quad \text{s.t.} \quad Z^v \geq 0,\ Z^{v\top}\mathbf{1} = \mathbf{1}, \tag{10}$$

where $\alpha > 0$ is a regularization parameter that prevents trivial solutions and promotes well-conditioned graphs. The non-negativity constraint $Z^v \geq 0$ ensures that the affinity values are interpretable as membership weights, and the column-sum constraint $Z^{v\top}\mathbf{1} = \mathbf{1}$ normalizes each sample's affinities to lie on the probability simplex. The entry Z^v_{jk} represents the reconstruction weight of the k-th sample with respect to the j-th anchor.

The reconstruction is performed in the projected space $\mathbb{R}^{d'_v}$ rather than in the original space $\mathbb{R}^{d_v}$. This distinction is essential when d_v is large, because the Frobenius norm in (10) measures distances through d'_v-dimensional inner products, yielding more reliable affinity estimates by concentrating on the variance-preserving directions.

3.2. Consensus Graph Fusion

Each view provides a different perspective of the underlying cluster structure, and the view-specific bipartite graphs $\{Z^v\}_{v=1}^V$ may exhibit both complementary patterns and view-specific artifacts. To aggregate cross-view information, a consensus bipartite graph P is introduced through the weighted discrepancy minimization

$$\min_{P \geq 0} \sum_{v=1}^{V} \omega_v \|P - Z^v\|_F^2, \tag{11}$$

where $\omega_v > 0$ denotes the adaptive weight for the v-th view. This formulation encourages P to serve as a centroid of the view-specific graphs in the Frobenius norm sense, with views of higher quality exerting greater influence through larger weights. The fusion objective in (11) treats all views with permutation symmetry, meaning that relabeling the view indices does not alter the optimization landscape. Unlike methods that fuse graphs at the spectral embedding level [19] or through direct concatenation [11], operating in the bipartite graph space preserves the $\mathcal{O}(Nr)$ complexity throughout the fusion process.

Different views contribute unequally to the clustering task due to varying feature quality and relevance to the underlying structure. To adaptively balance view contributions, each view is assigned a weight $\omega_v > 0$ subject to $\sum_{v=1}^{V} \omega_v = 1$, and an entropy regularization term is introduced as

$$\delta \sum_{v=1}^{V} \omega_v \ln \omega_v, \tag{12}$$

where $\delta > 0$ controls the smoothness of the weight distribution. A large δ drives the weights toward the uniform distribution $\omega_v = 1/V$, while a small δ concentrates weight on the views with the lowest reconstruction cost. The entropy term prevents degenerate solutions where all weight collapses onto a single view, and it admits a closed-form softmax solution as derived in Section 3.4. The temperature parameter is set adaptively as $\delta = \bar{h}/\ln(V+1)$, where $\bar{h} = (1/V)\sum_{v=1}^{V} h_v$ denotes the mean per-view cost, thereby scaling the entropy regularization to the magnitude of the objective without manual tuning.

To extract the cluster structure encoded in the consensus bipartite graph P, the augmented bipartite graph is formed as

$$S = \begin{bmatrix} 0 & P^\top \\ P & 0 \end{bmatrix} \in \mathbb{R}^{(N+r)\times(N+r)}. \tag{13}$$

The symmetric normalized Laplacian of S is $L_S = I - D_S^{-1/2} S D_S^{-1/2}$, where D_S denotes the diagonal degree matrix. By Lemma 1, the number of connected components in S equals the multiplicity of the zero eigenvalue of L_S. Let $L_P \in \mathbb{R}^{N\times N}$ denote the sample-node normalized Laplacian induced by the consensus bipartite graph P. It is used to compute the sample-level spectral embedding, while the augmented graph S is defined on both samples and anchors. A spectral embedding $\hat{F}$ is extracted from the c eigenvectors of L_P associated with the smallest eigenvalues, subject to the orthonormality constraint $\hat{F}^\top \hat{F} = I_c$. The spectral embedding term

$$\mathrm{Tr}\left(\hat{F}^\top L_P \hat{F}\right) \tag{14}$$

encourages P to possess a clear block-diagonal structure conducive to clustering.

To bridge the gap between the continuous embedding $\hat{F}$ and discrete cluster assignments, the spectral rotation framework [29] is adopted. An orthogonal matrix R is sought such that the rotated embedding $\hat{F}R$ aligns closely with a discrete indicator matrix $Y \in \mathrm{Ind}$. The alignment is measured by

$$\max_{R^\top R = I_c,\, Y \in \mathrm{Ind}} \mathrm{Tr}\left((\hat{F}R)^\top Y\right). \tag{15}$$

The orthogonal rotation R absorbs the inherent rotational ambiguity of the spectral embedding, and the indicator matrix Y provides a discrete cluster assignment through the row-wise argmax of $\hat{F}R$. This step avoids K-means post-processing during label refinement. In the experiments, the final labels are obtained by connected-component detection on

the converged consensus graph P, which reduces the dependence on K-means in the final assignment stage with fixed anchor initialization and parameter settings.

3.3. Objective Function

Given the fixed projected data and anchors, the complete optimization problem is

$$\begin{aligned}\min_{\substack{Z^v,P,\hat{F},\\ R,Y,\boldsymbol{\omega}}} \quad & \sum_{v=1}^{V}\omega_v\left\|\tilde{X}^{v\top}-\tilde{A}^{v\top}Z^v\right\|_F^2+\alpha\sum_{v=1}^{V}\omega_v\|Z^v\|_F^2+\beta\sum_{v=1}^{V}\omega_v\|P-Z^v\|_F^2\\ & +\operatorname{Tr}\left(\hat{F}^\top L_P\hat{F}\right)-\eta\operatorname{Tr}\left((\hat{F}R)^\top Y\right)+\delta\sum_{v=1}^{V}\omega_v\ln\omega_v\\ \text{s.t.}\quad & Z^v\geq 0,\ Z^{v\top}\mathbf{1}=\mathbf{1},\ P\geq 0,\ \hat{F}^\top\hat{F}=I_c,\ R^\top R=I_c,\ Y\in\text{Ind},\ \textstyle\sum_{v=1}^{V}\omega_v=1,\end{aligned}\tag{16}$$

where α, β, $\eta>0$ are trade-off parameters. The six terms in (16) serve complementary roles. The reconstruction term and the Frobenius regularization together measure anchor-based approximation quality in the projected subspace while preventing trivial solutions for Z^v. The consensus fusion term couples all views through the shared graph P, and the spectral embedding term extracts the block-diagonal structure of P via its symmetric normalized Laplacian L_P. The discrete rotation term aligns the continuous embedding to cluster indicators, and the entropy term adaptively distributes view weights.

3.4. Optimization

Problem (16) is solved by alternating minimization. Each variable is updated in turn while all others are held fixed, and every subproblem admits either a closed-form solution or an efficient iterative procedure. The update rules are presented below as steps (A) through (E).

(A) Update Z^v. With all other variables fixed, the positive weight ω_v does not affect the location of the minimizer and can be dropped. The subproblem for the v-th view is

$$\min_{Z^v\geq 0,\,Z^{v\top}\mathbf{1}=\mathbf{1}}\left\|\tilde{X}^{v\top}-\tilde{A}^{v\top}Z^v\right\|_F^2+\alpha\|Z^v\|_F^2+\beta\|P-Z^v\|_F^2.\tag{17}$$

Expanding each term using $\|M\|_F^2=\operatorname{Tr}(M^\top M)$ yields

$$\left\|\tilde{X}^{v\top}-\tilde{A}^{v\top}Z^v\right\|_F^2=\operatorname{Tr}\left(\tilde{X}^v\tilde{X}^{v\top}\right)-2\operatorname{Tr}\left(Z^v\tilde{X}^v\tilde{A}^{v\top}\right)+\operatorname{Tr}\left(Z^{v\top}\tilde{A}^v\tilde{A}^{v\top}Z^v\right),\tag{18}$$

$$\alpha\|Z^v\|_F^2=\alpha\operatorname{Tr}\left(Z^vZ^{v\top}\right),\tag{19}$$

$$\beta\|P-Z^v\|_F^2=\beta\operatorname{Tr}(PP^\top)-2\beta\operatorname{Tr}(PZ^{v\top})+\beta\operatorname{Tr}(Z^vZ^{v\top}).\tag{20}$$

Collecting all terms that depend on Z^v and discarding constants, the objective reduces to

$$\operatorname{Tr}\left(\tilde{A}^v\tilde{A}^{v\top}Z^vZ^{v\top}\right)+(\alpha+\beta)\operatorname{Tr}\left(Z^vZ^{v\top}\right)-2\operatorname{Tr}\left(Z^v\tilde{X}^v\tilde{A}^{v\top}\right)-2\beta\operatorname{Tr}\left(PZ^{v\top}\right).\tag{21}$$

Taking the matrix derivative with respect to Z^v and applying the identities

$$\frac{\partial\operatorname{Tr}(Z^\top AZ)}{\partial Z}=2AZ,\quad \frac{\partial\operatorname{Tr}(BZ^\top)}{\partial Z}=B,\tag{22}$$

where A is symmetric, the stationarity condition becomes

$$2\,\tilde{A}^v\tilde{A}^{v\top}Z^v+2(\alpha+\beta)\,Z^v-2\,\tilde{A}^v\tilde{X}^{v\top}-2\beta\,P=0.\tag{23}$$

Dividing by 2 and rearranging,

$$\left[\tilde{A}^v \tilde{A}^{v\top} + (\alpha + \beta)\, I_r\right] Z^v = \tilde{A}^v \tilde{X}^{v\top} + \beta P. \tag{24}$$

The coefficient matrix

$$\Phi^v = \tilde{A}^v \tilde{A}^{v\top} + (\alpha + \beta)\, I_r \tag{25}$$

is symmetric positive definite, since $\tilde{A}^v \tilde{A}^{v\top}$ is positive semidefinite and $\alpha + \beta > 0$. The unconstrained minimizer is therefore unique and given by

$$\hat{Z}^v = (\Phi^v)^{-1}\left[\tilde{A}^v \tilde{X}^{v\top} + \beta P\right]. \tag{26}$$

The matrix Φ^v is of size $r \times r$ with $r \ll N$, so its Cholesky factorization costs $\mathcal{O}(r^3)$ and the subsequent back-substitution costs $\mathcal{O}(r^2 N)$. The dominant cost is the product $\tilde{A}^v \tilde{X}^{v\top}$, which requires $\mathcal{O}(r d'_v N)$ operations. The regularization term βP in the right-hand side of (26) draws each view-specific graph toward the shared consensus graph, with the strength of this effect governed by β.

The unconstrained solution $\hat{Z}^v$ generally violates the constraints $Z^v \geq 0$ and $Z^{v\top}\mathbf{1} = \mathbf{1}$. Each column of $\hat{Z}^v$ is projected onto the probability simplex, yielding $z_i^* = \max(y_i - \theta, 0)$, where θ is determined in $\mathcal{O}(r \log r)$ time [31].

(B) Update *P*. With all other variables fixed, the terms involving P are

$$\min_{P \geq 0} \beta \sum_{v=1}^{V} \omega_v \|P - Z^v\|_F^2 + \mathrm{Tr}\left(\hat{F}^\top L_P \hat{F}\right). \tag{27}$$

A two-stage approach is adopted. The weighted-average step gives the exact minimizer of the quadratic consensus fitting term, while the subsequent coclustering refinement enforces the desired bipartite graph structure. Therefore, the overall P-update is interpreted as an approximate block update for the full P-subproblem rather than a simultaneous closed-form solution to all terms involving P. The fusion term is minimized by expanding $\sum_v \omega_v \|P - Z^v\|_F^2$ and differentiating with respect to P,

$$2\left(\sum_{v=1}^{V} \omega_v\right) P - 2 \sum_{v=1}^{V} \omega_v Z^v = 0, \tag{28}$$

which yields the weighted average

$$\bar{Z} = \frac{\sum_{v=1}^{V} \omega_v Z^v}{\sum_{v=1}^{V} \omega_v}, \tag{29}$$

followed by projection onto the non-negative orthant. The spectral term is then incorporated through coclustering refinement [12,13]. The augmented graph S is formed as in (13), and its symmetric normalized Laplacian is $L_S = I - D_S^{-1/2} S\, D_S^{-1/2}$. The refinement is applied to the bipartite block rather than to the whole augmented adjacency matrix. Specifically, it solves

$$\min_{P \geq 0,\, P^\top \mathbf{1} = \mathbf{1}} \|P - \bar{Z}\|_F^2 + 2\lambda\, \mathrm{Tr}\left(F_S^\top L_{S(P)} F_S\right), \quad \text{s.t.} \quad F_S^\top F_S = I_c \tag{30}$$

where $S(P)$ denotes the augmented bipartite adjacency matrix constructed from the current bipartite block P as in Equation (13), $L_{S(P)}$ is its symmetric normalized Laplacian, and $F_S \in \mathbb{R}^{(N+r)\times c}$ is the spectral embedding associated with the augmented graph. Following the adaptive strategy of [13,16], λ is doubled when the number of near-zero eigenvalues of $L_{S(P)}$ is less than c and halved when it exceeds c, until the augmented graph possesses

exactly c connected components. This mechanism eliminates a hyperparameter while imposing the desired cluster count.

(C) Update $\hat{F}$. With all other variables fixed, the subproblem is

$$\min_{\hat{F}^\top \hat{F}=I_c} \operatorname{Tr}\Big(\hat{F}^\top L_P \hat{F}\Big) - \eta \operatorname{Tr}\Big((\hat{F}R)^\top Y\Big), \tag{31}$$

which is equivalently written as

$$\max_{\hat{F}^\top \hat{F}=I_c} \operatorname{Tr}\Big[\hat{F}^\top \Big(-L_P \hat{F} + \eta Y R^\top\Big)\Big]. \tag{32}$$

Following Luo et al. [20], this is solved via iterative SVD. At the k-th inner step, form

$$N_k = -2L_P \hat{F}_k + \eta Y R^\top, \tag{33}$$

compute its thin SVD $N_k = U_k \Sigma_k V_k^\top$, and update

$$\hat{F}_{k+1} = U_k V_k^\top. \tag{34}$$

By the von Neumann trace inequality, the update $\hat{F}_{k+1} = U_k V_k^\top$ maximizes $\operatorname{Tr}(\hat{F}^\top N_k)$ over all matrices with orthonormal columns. This provides an efficient orthogonality-preserving update for the inner SVD step. The embedding $\hat{F}$ is initialized from the spectral decomposition of P and evolves smoothly across outer iterations without re-initialization; two to three inner iterations suffice in practice.

(D) Update R and Y. With all other variables fixed, the subproblem for R reduces to the orthogonal Procrustes problem

$$\max_{R^\top R=I_c} \operatorname{Tr}\Big(R^\top \hat{F}^\top Y\Big). \tag{35}$$

Let $\hat{F}^\top Y = U_R \Sigma_R V_R^\top$ be the thin SVD. By the von Neumann trace inequality

$$\operatorname{Tr}(R^\top \hat{F}^\top Y) = \operatorname{Tr}(R^\top U_R \Sigma_R V_R^\top) \le \operatorname{Tr}(\Sigma_R), \tag{36}$$

and the upper bound is attained at

$$R^* = U_R V_R^\top. \tag{37}$$

For the indicator matrix Y, the subproblem with R and $\hat{F}$ fixed is

$$\max_{Y \in \mathrm{Ind}} \operatorname{Tr}\Big(Y^\top \hat{F} R\Big). \tag{38}$$

Let $F^* = \hat{F}R$. Since Y is an indicator matrix, the trace decomposes row-wise as $\operatorname{Tr}(Y^\top F^*) = \sum_{i=1}^{N} F^*_{i,c_i}$, where c_i indexes the unit entry in the i-th row of Y. Each row is independent, and the global maximum is achieved by

$$Y_{ij} = \begin{cases} 1 & \text{if } j = \arg\max_k \ F^*_{ik}, \\ 0 & \text{otherwise.} \end{cases} \tag{39}$$

(E) Update ω. Collecting all terms involving ω_v from (16), the subproblem is

$$\min_{\omega_v>0,\sum_v \omega_v=1} \sum_{v=1}^{V} \omega_v h_v + \delta \sum_{v=1}^{V} \omega_v \ln \omega_v, \tag{40}$$

where the per-view cost aggregating the reconstruction, regularization, and consensus terms of (16) is

$$h_v = \left\| \tilde{X}^{v\top} - \tilde{A}^{v\top} Z^v \right\|_F^2 + \alpha \|Z^v\|_F^2 + \beta \|P - Z^v\|_F^2. \tag{41}$$

Introducing a Lagrange multiplier μ for the constraint $\sum_v \omega_v = 1$, the Lagrangian is

$$\mathcal{L}(\boldsymbol{\omega}, \mu) = \sum_{v=1}^{V} \omega_v h_v + \delta \sum_{v=1}^{V} \omega_v \ln \omega_v - \mu \left(\sum_{v=1}^{V} \omega_v - 1 \right). \tag{42}$$

Setting $\partial \mathcal{L} / \partial \omega_v = 0$ gives

$$h_v + \delta(\ln \omega_v + 1) - \mu = 0, \tag{43}$$

from which

$$\omega_v = \exp\left(\frac{\mu - \delta - h_v}{\delta} \right) = \exp\left(\frac{\mu - \delta}{\delta} \right) \exp\left(\frac{-h_v}{\delta} \right). \tag{44}$$

Substituting into $\sum_v \omega_v = 1$ to eliminate μ yields

$$\exp\left(\frac{\mu - \delta}{\delta} \right) = \left[\sum_{k=1}^{V} \exp\left(\frac{-h_k}{\delta} \right) \right]^{-1}. \tag{45}$$

Substituting back into (44) gives the softmax solution

$$\omega_v = \frac{\exp(-h_v/\delta)}{\sum_{k=1}^{V} \exp(-h_k/\delta)}. \tag{46}$$

The objective (40) is strictly convex in $\boldsymbol{\omega}$ for any $\delta > 0$ because its Hessian is $\delta \cdot \mathrm{diag}(1/\omega_1, \ldots, 1/\omega_V)$, which is positive definite. Therefore (46) is the unique global minimizer.

3.5. Algorithm Summary

The complete procedure is summarized in Algorithm 1.

Algorithm 1 PEBGL: projection-enhanced bipartite graph learning.

Require: Multi-view data $\{X^v\}_{v=1}^V$, number of clusters c, anchor count r, parameters α, β, η.

1: Generate joint anchors $\{A^v\}$ via concatenation and K-means with 30 restarts.
2: Compute PCA projections W^v once and keep them fixed; form $\tilde{X}^v = X^v W^v$ and $\tilde{A}^v = A^v W^v$.
3: Initialize Z^v via distance-based simplex projection.
4: Initialize $P = (1/V) \sum_v Z^v$.
5: Initialize $\hat{F}$ from the spectral embedding of P.
6: Initialize Y via K-means on $\hat{F}$; set $R = I_c$; set $\omega_v = 1/V$.
7: **while** not converged **do**
8: Update Z^v for each view via Equation (26) and simplex projection.
9: Update P via Equation (29) and coclustering refinement (30).
10: Update $\hat{F}$ via Equations (33) and (34) with 2–3 inner SVD iterations.
11: Update R via Equation (37).
12: Update Y via Equation (39).
13: Update $\boldsymbol{\omega}$ via Equation (46) with $\delta = \bar{h}/\ln(V+1)$.
14: **end while**

Ensure: Cluster labels from connected-component detection on P.

The stopping criterion is defined by the relative change in the objective function. Specifically, the iteration terminates when $\frac{|J^t - J^{t-1}|}{|J^{t-1}| + 10^{-10}} < 10^{-5}$, or when the maximum number of iterations reaches 50, where J^t denotes the objective value at iteration t.

Remark 1. *The indicator matrix Y is initialized via K-means applied to the initial spectral embedding $\hat{F}$. Within the main loop, Y is updated exclusively by the row-wise argmax rule (39), and no further K-means invocation occurs for label refinement. Although Y provides a discrete indicator during optimization, the experiments in Section 4 report results from connected-component detection on the converged consensus graph P. Therefore, PEBGL reduces the dependence on K-means post-processing in the final label assignment stage, and the final decoding step is reproducible with fixed anchor initialization, random seed, and parameter settings.*

3.6. Complexity

The per-iteration cost of Algorithm 1 is analyzed as follows. Updating Z^v for a single view requires $\mathcal{O}(rd'_v N)$ operations, dominated by the matrix product $\tilde{A}^v \tilde{X}^{v\top}$. The Cholesky factorization of the $r \times r$ matrix Φ^v contributes $\mathcal{O}(r^3)$, and the column-wise simplex projection contributes $\mathcal{O}(Nr \log r)$. Summing over V views gives $\mathcal{O}(Vrd'_{\max} N + Vr^3 + VNr \log r)$.

The consensus graph update via the weighted average and coclustering refinement costs $\mathcal{O}((V + T_P)rN)$, where T_P is the number of inner refinement steps. The spectral embedding, rotation, discrete label update, and weight update collectively cost $\mathcal{O}(Nc^2 + c^3 + V)$, where the additional terms are usually small because $c \ll N$ in typical clustering tasks. Therefore, when the numbers of views, anchors, projected dimensions, clusters, and inner refinement steps are fixed, the dominant iterative graph-learning cost scales linearly with the number of samples N.

The PCA projection is performed once before the main alternating optimization. Its one-time preprocessing cost is $\mathcal{O}(\sum_{v=1}^{V} d_v^2 N)$ when using a standard PCA implementation. This cost is outside the main iterative loop and may become non-negligible for high-dimensional small-sample datasets, which is consistent with the runtime behavior observed in Section 4.7.

4. Experiments

All experiments were executed in MATLAB R2024a on a machine equipped with an AMD Ryzen 9 7945HX processor and 32 GB of RAM.

4.1. Datasets and Evaluation Metrics

Six publicly available multi-view datasets spanning image recognition, handwritten digit classification, and plant species identification were adopted. Table 2 summarizes their statistics.

Table 2. Descriptions of datasets.

Dataset	N	V	c	d_1	d_2	d_3	d_4	d_5	d_6
MSRCV1	210	5	7	24	576	512	256	254	–
Yale	165	3	15	4096	3304	6750	–	–	–
ORL	400	3	40	4096	3304	6750	–	–	–
Handwritten	2000	6	10	76	216	64	240	47	6
100Leaves	1600	3	100	64	64	64	–	–	–
Caltech101-7	1474	6	7	48	40	254	1984	512	928

Here N, V, and c denote the number of samples, views, and clusters, respectively, and d_v is the feature dimension of the v-th view. "–" indicates that the corresponding view does not exist.

MSRCV1 [34] contains 210 images from 7 semantic categories, each described by 5 feature types, namely, color moments, HOG descriptors, GIST features, LBP textures, and CENTRIST features.

Yale [35] consists of 165 face images of 15 individuals captured with varying illumination and expressions, represented by 3 high-dimensional views.

ORL [36] includes 400 face images from 40 subjects with variations in lighting, expression, and facial details, also described by 3 views with the same feature types as Yale.

Handwritten [37] contains 2000 images of handwritten digits 0 through 9, characterized by 6 feature types, namely, Fourier coefficients, profile correlations, Karhunen–Loève coefficients, pixel averages, Zernike moments, and morphological features.

100Leaves [38] comprises 1600 plant leaf samples from 100 species described by 3 views of shape, fine-scale margin, and texture descriptors, all with 64-dimensional features.

Caltech101-7 [39] contains 1474 images from 7 object categories, described by 6 views, namely, Gabor features, wavelet moments, CENTRIST features, HOG descriptors, GIST features, and LBP textures.

Three widely used external evaluation metrics were adopted. Accuracy [40] measures the proportion of correctly assigned samples after optimal label permutation via the Hungarian algorithm. Normalized Mutual Information [41] quantifies the mutual dependence between predicted and true labels, normalized to the range $[0, 1]$. Purity [42] computes the fraction of samples in each cluster that belong to the dominant class. Higher values indicate better clustering quality for all three metrics.

4.2. Experimental Setup

PEBGL was compared against seven recent multi-view clustering methods based on anchor graphs or bipartite graph learning. GFSC [43] performs multi-graph fusion for multi-view spectral clustering. LMVSC [11] constructs anchor graphs per view independently and fuses them through concatenation. CDMGC [9] decomposes view graphs into consistent and diverse components for fusion. MSGL [15] develops structured graph learning from single-view to multi-view settings. SFMC [13] learns a parameter-free consensus bipartite graph with a Laplacian rank constraint. FPMVS-CAG [14] integrates anchor selection and graph construction without explicit hyperparameter tuning. DiBGF-MGC [18] separates bipartite graphs into consistency and diversity components through intra-view and inter-view constraints.

Although SFMC, FPMVS-CAG, DiBGF-MGC, and PEBGL all belong to scalable graph-based or bipartite graph-based multi-view clustering methods, their technical emphases are different. SFMC learns a parameter-free consensus bipartite graph under a Laplacian rank constraint, but sample-anchor affinities are still constructed in the original feature space. FPMVS-CAG integrates anchor selection and graph construction through consensus anchor guidance, while the learned affinities also rely on the original feature representation. DiBGF-MGC improves graph fusion by decomposing bipartite graphs into consistency and diversity components. In contrast, PEBGL applies fixed PCA-based projection preprocessing before sample-anchor affinity learning, constructs projection-enhanced bipartite graphs, performs entropy-regularized adaptive consensus fusion, and obtains final labels by connected-component detection without K-means post-processing. Therefore, the main difference of PEBGL lies in projection-enhanced graph construction, adaptive consensus fusion, and K-means-free final decoding within a scalable bipartite graph framework.

For the compared methods, publicly available implementations were used whenever available, and the same unsupervised evaluation protocol was followed on the benchmark datasets. When complete implementations or identical preprocessing settings were not fully available, the results reported in the corresponding papers were used as references.

Available implementations were also tested in the same local MATLAB environment for the execution-time comparison in Section 4.7. For PEBGL, anchors were initialized by K-means with 30 restarts with a fixed random seed, while the final labels were obtained by connected-component detection rather than K-means post-processing.

For PEBGL, the hyperparameters were tuned via a two-stage grid search. In the first stage, α and β were swept over logarithmic grids $\{10^{-3}, \ldots, 10^{3}\}$ and $\{10^{-3}, \ldots, 10^{2}\}$ respectively at each candidate r to identify the promising region. In the second stage, a finer search was performed around the identified optimum. The anchor count r was selected from a dataset-dependent candidate set ranging from c to $10c$. The rotation trade-off was fixed at $\eta = 1$. The entropy temperature δ was set adaptively as $\delta = \bar{h}/\ln(V+1)$, where $\bar{h}$ denotes the mean reconstruction cost across views. For high-dimensional small-sample datasets where $d_v \gg N$, the PCA projection dimension was capped at c to avoid having fewer samples than features after projection; this applies to ORL in the present experiments.

4.3. Clustering Results

Tables 3–5 present the clustering results measured by ACC, NMI, and Purity on six benchmark datasets. The best result in each row is highlighted in **bold**, and the second best is underlined.

Table 3. ACC (%) on six datasets.

Dataset	GFSC	LMVSC	CDMGC	MSGL	SFMC	FPMVS-CAG	DiBGF-MGC	PEBGL
100Leaves	39.9	61.4	86.3	48.8	70.9	35.6	87.2	**88.5**
Caltech101-7	49.3	72.7	73.6	73.3	65.3	61.5	80.1	**84.9**
MSRCV1	71.4	77.6	69.1	72.4	81.0	60.5	**83.3**	81.9
Handwritten	70.8	91.7	98.8	74.4	97.9	82.3	**99.1**	96.9
Yale	51.4	61.3	69.5	15.8	58.8	44.9	**75.6**	71.5
ORL	56.4	60.6	79.0	25.3	61.5	56.0	**79.2**	70.8

Table 4. NMI (%) on six datasets.

Dataset	GFSC	LMVSC	CDMGC	MSGL	SFMC	FPMVS-CAG	DiBGF-MGC	PEBGL
100Leaves	71.0	80.7	92.4	56.2	83.0	70.3	93.9	**94.2**
Caltech101-7	2.9	51.9	52.5	52.5	55.5	57.0	62.1	**64.8**
MSRCV1	64.6	66.9	64.3	57.3	72.1	55.6	**75.7**	74.5
Handwritten	68.9	84.4	97.2	75.1	94.8	79.2	**98.0**	93.3
Yale	55.1	79.7	68.9	12.7	60.0	51.6	**85.9**	70.6
ORL	72.4	78.1	84.1	45.9	76.6	76.3	**91.5**	83.8

Table 5. Purity (%) on six datasets.

Dataset	GFSC	LMVSC	CDMGC	MSGL	SFMC	FPMVS-CAG	DiBGF-MGC	PEBGL
100Leaves	79.3	70.1	88.3	60.0	72.8	36.9	89.4	**90.1**
Caltech101-7	**92.8**	75.2	89.1	77.3	85.3	86.6	90.5	89.4
MSRCV1	74.0	77.6	70.0	76.7	81.0	61.9	**84.5**	81.9
Handwritten	78.2	91.7	98.8	88.1	97.9	82.3	**99.1**	96.9
Yale	60.7	71.3	70.3	66.1	60.0	47.3	**80.5**	71.5
ORL	72.0	70.5	76.0	43.3	68.0	60.0	**88.8**	73.5

Tables 3–5 show that PEBGL attains the highest ACC and NMI on 100Leaves and Caltech101-7, two datasets that differ substantially in structure. The 100Leaves dataset includes 100 fine-grained plant species with low-dimensional views, while Caltech101-7 combines six feature types with dimensionality spanning from 40 to 1984. On 100Leaves,

PEBGL obtains slightly higher results than DiBGF-MGC by 1.3, 0.3, and 0.7 percentage points in ACC, NMI, and Purity, respectively. On Caltech101-7, PEBGL improves ACC and NMI by 4.8 and 2.7 percentage points, respectively.

On MSRCV1, PEBGL ranks second across all three metrics, trailing DiBGF-MGC by only 1.4 points in ACC while outperforming the remaining baselines by clear margins. On Handwritten, all top methods achieve above 95% ACC; PEBGL reaches 96.9%, close to DiBGF-MGC and CDMGC. The small gap reflects the well-separated digit classes and balanced six-view configuration. On Yale and ORL, PEBGL achieves competitive but not leading results compared with DiBGF-MGC. Both datasets are high-dimensional small-sample face recognition benchmarks, where the largest view dimension reaches 6750, while the sample sizes are only 165 and 400, respectively. With this setting, the fixed PCA projection preserves dominant variance directions but may weaken some low-variance facial cues related to subtle illumination, expression, and local texture variations. In contrast, DiBGF-MGC constructs bipartite graphs in the original feature space and separates consistency and diversity components, which may better preserve view-specific facial structures.

In summary, across the 18 metric–dataset combinations, PEBGL achieves the best result five times and the second-best result five times. These results indicate that the proposed framework achieves competitive clustering accuracy on datasets with diverse scales and feature configurations.

4.4. Ablation Study

To evaluate the contribution of each module, four ablation variants were constructed. PEBGL-1 removes the PCA projection and constructs bipartite graphs directly in the original feature space. PEBGL-2 sets $\beta = 0$ so that each view-specific bipartite graph is learned independently without consensus feedback. PEBGL-3 replaces the connected-component decoding with K-means applied to the spectral embedding of the converged P. PEBGL-4 fixes $\omega_v = 1/V$ throughout the optimization, disabling adaptive view weighting. Table 6 reports the ACC of all variants on six datasets. Bold indicates the best result.

Table 6. Ablation study measured by ACC (%).

Variant	100Leaves	Caltech101-7	MSRCV1	Handwritten	Yale	ORL
PEBGL-1	88.5	42.9	64.8	84.9	16.4	70.3
PEBGL-2	42.1	50.9	72.4	68.9	67.9	61.5
PEBGL-3	68.2	53.3	77.6	94.9	66.1	62.0
PEBGL-4	86.2	47.3	76.2	96.4	66.7	70.3
PEBGL	**88.5**	**84.9**	**81.9**	**96.9**	**71.5**	**70.8**

PEBGL-1 removes the fixed PCA projection and constructs bipartite graphs directly in the original feature space. This causes the largest drop on high-dimensional datasets, with ACC falling from 71.5% to 16.4% on Yale and from 84.9% to 42.9% on Caltech101-7, indicating that the projection step is effective in reducing redundant high-dimensional information before sample-anchor graph construction. On 100Leaves, all three views are 64-dimensional, so the projection has no measurable effect. It should also be noted that the PCA projection is fixed and unsupervised and thus may not fully preserve all cluster-discriminative components on high-dimensional small-sample face datasets. PEBGL-2 disables consensus fusion and yields the largest average degradation across all six datasets. PEBGL-3 replaces connected-component decoding with K-means, reducing ACC on every dataset. PEBGL-4 fixes uniform view weights and suffers the most on Caltech101-7, where the six views differ greatly in quality. Overall, consensus fusion contributes the most,

followed by the decoding strategy and adaptive weighting, while projection is decisive only for high-dimensional views.

4.5. Parameter Analysis

The sensitivity of PEBGL to its two primary hyperparameters α and β is investigated by varying each over $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3\}$ and $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2\}$ respectively while fixing r at its optimal value. Figure 2 visualizes ACC as 3D bar charts on six datasets.

The effect of the anchor count r on clustering accuracy is examined separately. With α and β fixed at their optimal values, r is varied over $\{c, 2c, 3c, 5c, 7c, 10c\}$. Figure 3 reports the results.

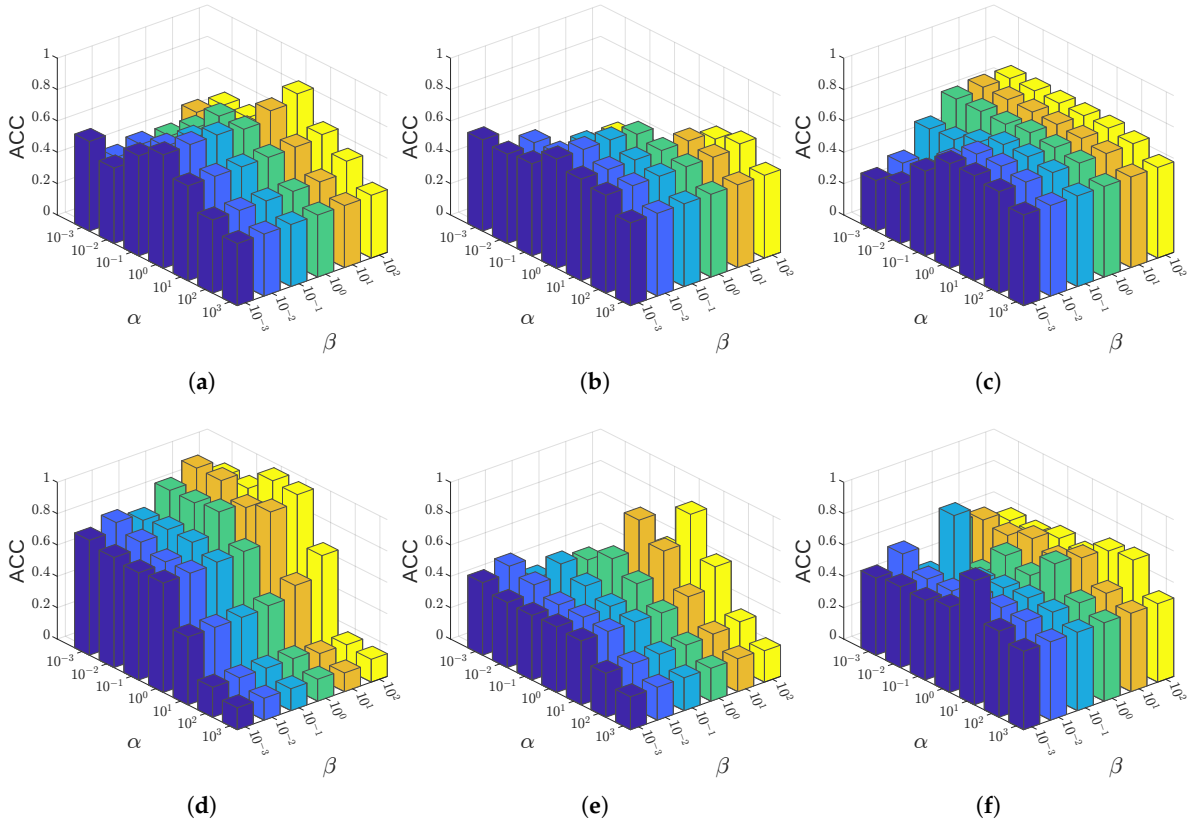


Figure 2. Parameter sensitivity of ACC with respect to α and β on six datasets. (**a**) MSRCV1. (**b**) Yale. (**c**) ORL. (**d**) Handwritten. (**e**) 100Leaves. (**f**) Caltech101-7.

As shown in Figure 2, PEBGL maintains stable ACC over a wide range of α and β on most datasets. On MSRCV1 and Handwritten, ACC remains above 60% and 80% respectively across several orders of magnitude, with degradation confined to extreme corners of the grid. Caltech101-7 shows a narrower optimal region concentrated at small α and β, and performance drops rapidly outside this region. Yale and ORL exhibit moderate sensitivity, with higher ACC values concentrated along intermediate α.

Figure 3 reveals that the optimal anchor count varies across datasets and does not follow a uniform trend. On Handwritten, ACC increases steadily with r and stabilizes near $r = 80$. On 100Leaves, performance peaks near $r = 900$, a value substantially larger than the class count $c = 100$, reflecting the need for a large anchor set to represent 100 leaf species. On Yale, ORL, and Caltech101-7, ACC peaks at a specific r and declines when r grows further, suggesting that excessive anchors introduce redundancy rather than additional discriminative information. On MSRCV1 the pattern is similar, with the best ACC achieved at $r = 45$.

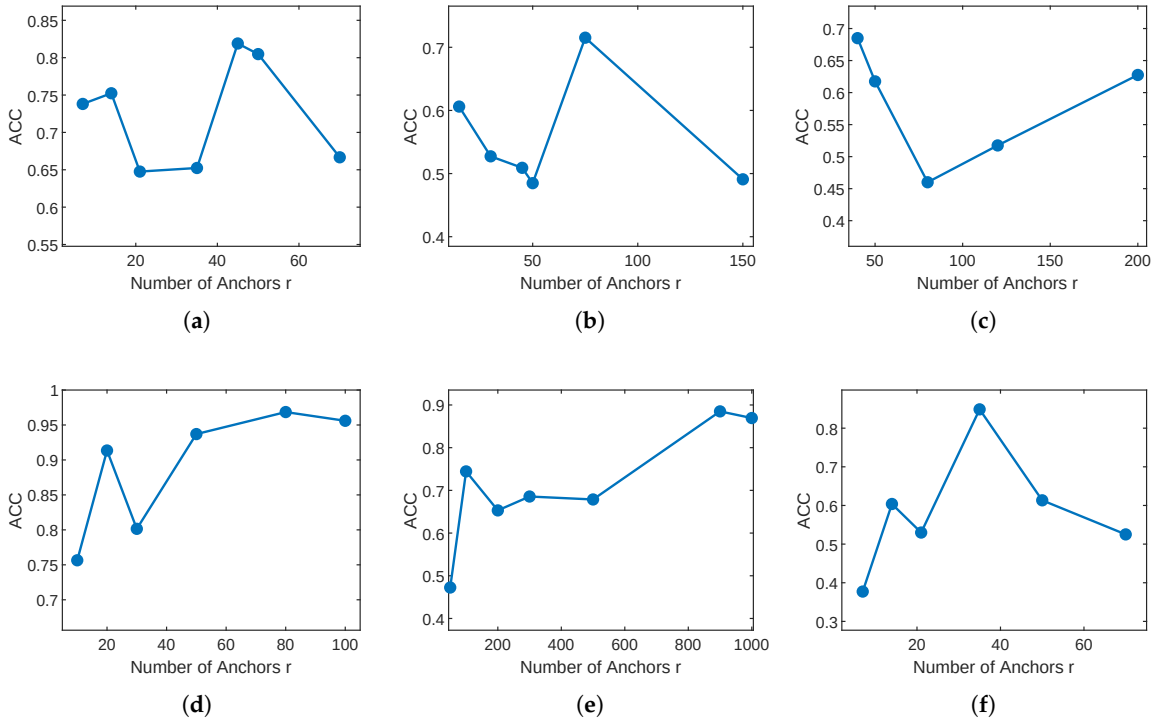


Figure 3. Effect of anchor count r on clustering ACC. (**a**) MSRCV1. (**b**) Yale. (**c**) ORL. (**d**) Handwritten. (**e**) 100Leaves. (**f**) Caltech101-7.

4.6. Behavior of the Consensus Graph

To examine how the consensus bipartite graph evolves during optimization, Figure 4 tracks the relative change $\|P^t - P^{t-1}\|_F / \|P^{t-1}\|_F$ across iterations on all six datasets.

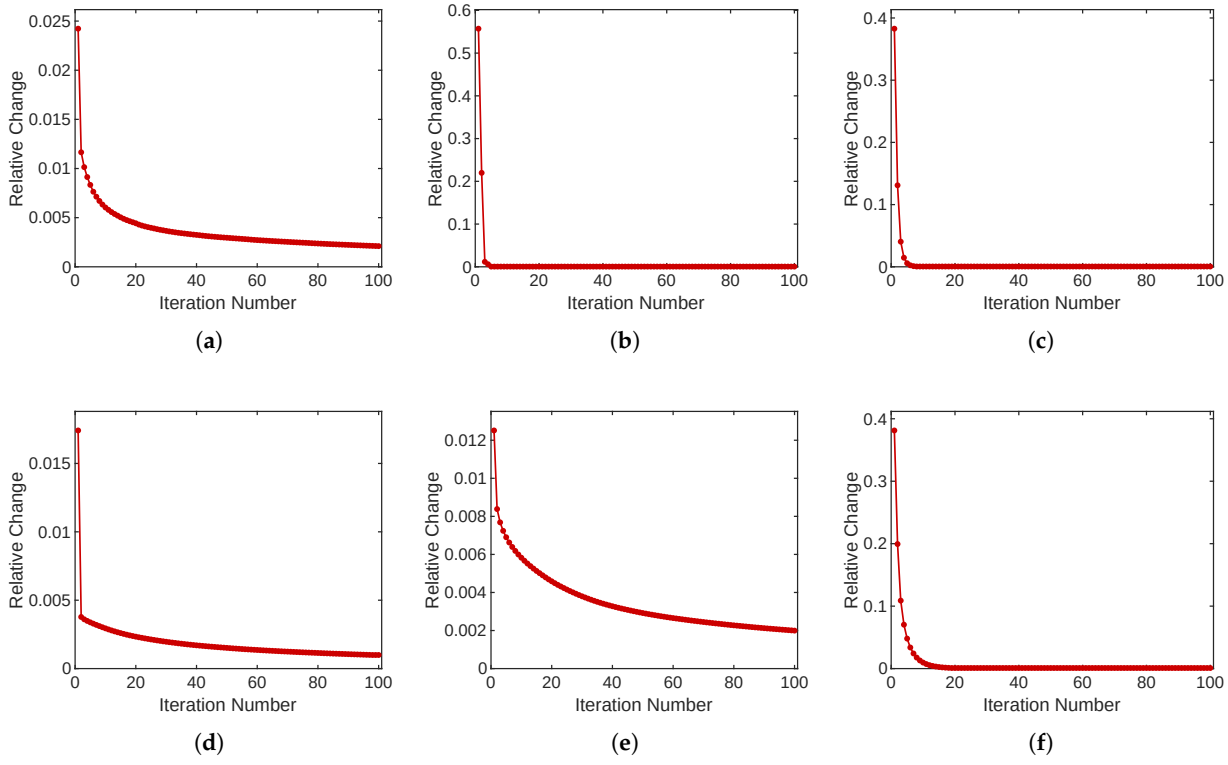


Figure 4. Relative change in the consensus graph P on six datasets. (**a**) MSRCV1. (**b**) Yale. (**c**) ORL. (**d**) Handwritten. (**e**) 100Leaves. (**f**) Caltech101-7.

As shown in Figure 4, the six datasets exhibit two distinct patterns. Yale, ORL, and Caltech101-7 start from relatively large initial changes above 0.3 and drop sharply within the first 10 iterations, after which the update magnitude remains close to zero. MSRCV1, Handwritten, and 100Leaves begin with much smaller initial changes below 0.03 and decrease gradually over the 100-iteration horizon. This difference is mainly related to the initialization quality of the consensus graph P, which already approximates a stable structure on the latter three datasets. In all cases, the relative change decreases to a negligible level, indicating stable empirical convergence behavior of the proposed alternating optimization procedure.

4.7. Computational Cost

Figure 5 reports the execution-time comparison of PEBGL and seven representative multi-view clustering methods on six benchmark datasets. The results show that PEBGL achieves competitive computational efficiency on several datasets. Specifically, PEBGL runs efficiently on MSRCV1, Handwritten, and Caltech101-7, and its running time is comparable to or lower than several recent graph-based and bipartite graph-based methods.

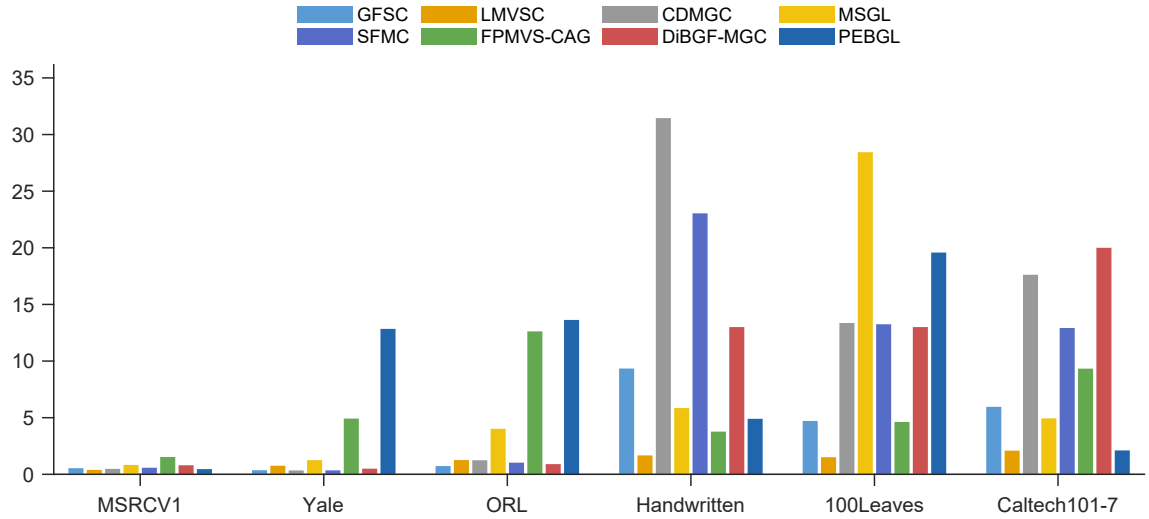


Figure 5. Execution-time (s) comparison of different methods on six benchmark datasets.

On Yale and ORL, PEBGL takes more time than most compared methods. This is mainly because these two face datasets contain very high-dimensional views, with the largest feature dimension reaching 6750, so the PCA preprocessing stage introduces additional computational cost before bipartite graph learning. On 100Leaves, the relatively high running time is mainly related to the large number of categories, where a larger anchor number is required to represent fine-grained leaf structures. Nevertheless, after the projection stage, the main iterative bipartite graph learning procedure remains efficient, which is consistent with the complexity analysis in Section 3.6.

4.8. View Weight Analysis

To illustrate the behavior of the entropy-regularized adaptive weighting mechanism, Figure 6 traces the evolution of the view weights ω_v across iterations on MSRCV1 and Handwritten, the two datasets with the largest number of views.

On MSRCV1, Color moment, LBP, and CENTRIST collectively receive over 96% of the total weight at the final iteration, while HOG and GIST are assigned weights below 0.02. This allocation reflects the feature characteristics of the dataset as HOG and GIST retain only 10 and 19 PCA dimensions after projection, suggesting limited discriminative capacity

in these two views. The weight distribution stabilizes within approximately 15 iterations and shifts only slightly thereafter.

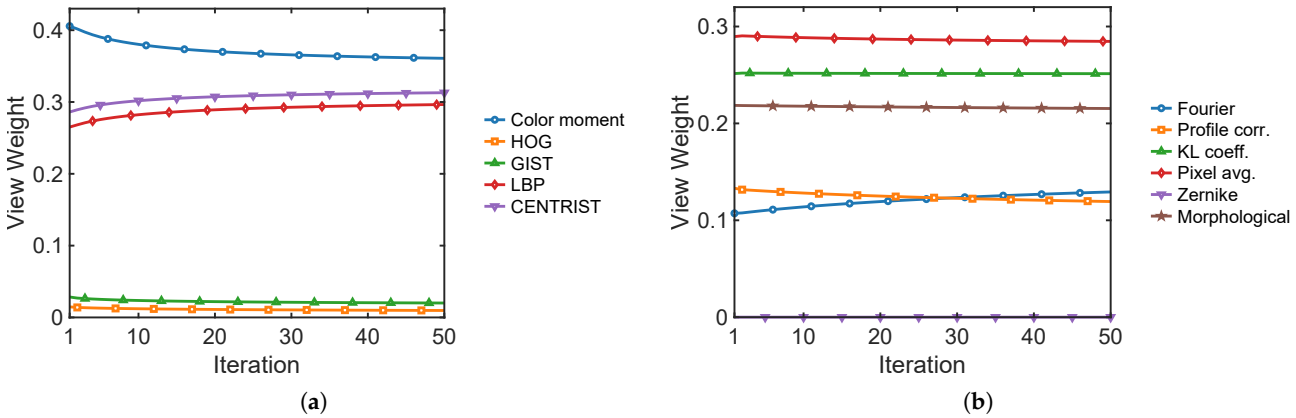


Figure 6. Evolution of adaptive view weights on MSRCV1 and Handwritten. The legends show the corresponding feature types of each dataset. (**a**) MSRCV1. (**b**) Handwritten.

On Handwritten, five of the six views share comparable weights between 0.12 and 0.30, whereas the Zernike view is effectively suppressed with $\omega \approx 3 \times 10^{-5}$. The 47-dimensional Zernike moment feature provides negligible complementary information beyond what the remaining five feature types already capture. In effect, the adaptive weighting acts as a soft view selection mechanism that concentrates the fusion on informative views without manual tuning. The entropy regularization prevents the weights from collapsing to a single-view solution, preserving meaningful contributions from multiple views.

5. Conclusions

This paper proposes PEBGL, a projection-enhanced bipartite graph learning framework for multi-view clustering. The method integrates fixed PCA-based projection preprocessing, bipartite graph construction, consensus fusion with entropy-penalized adaptive weighting, spectral embedding, and discrete rotation into an optimization framework. Before building the bipartite graph, each view is projected onto a compact subspace that preserves the dominant variance, which helps reduce redundant high-dimensional information and alleviate the distance concentration effect. The entropy regularization mechanism adjusts view weights automatically during optimization. The final cluster labels are obtained by detecting connected components on the converged consensus graph, reducing the dependence on K-means post-processing. The optimization subproblems admit efficient closed-form or projection-based updates, making the main graph-learning procedure computationally tractable.

Experiments on six benchmark datasets against seven recent methods show that PEBGL achieves competitive performance, obtaining the best accuracy on two datasets and comparable results on the remaining datasets. In future work, more flexible projection strategies will be considered to further improve the adaptability of PEBGL on high-dimensional and complex multi-view data.

Author Contributions: Conceptualization, X.L. and J.-F.C.; methodology, X.L. and J.-F.C.; software, X.L. and J.-F.C.; validation, X.L. and J.-F.C.; formal analysis, X.L.; resources, Q.-W.W.; data curation, X.L.; writing—original draft preparation, Q.-W.W. and X.L.; writing—review and editing, Q.-W.W., X.L. and J.-F.C.; visualization, X.L. and J.-F.C.; supervision, Q.-W.W.; project administration, Q.-W.W.; funding acquisition, Q.-W.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by the National Natural Science Foundation of China (No. 12371023).

Data Availability Statement: The datasets analyzed in this study are publicly available and can be accessed at: https://github.com/ChuanbinZhang/Multi-view-datasets (accessed on 10 May 2026).

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

References

1. Hamam, H. Rethinking intelligence: From human cognition to artificial futures. *Vokasi Unesa Bull. Eng. Technol. Appl. Sci.* **2025**, *2*, 531–548. [CrossRef]
2. Hamadneh, T.; Batiha, B.; Gharib, G.M.; Montazeri, Z.; Dehghani, M.; Aribowo, W.; Abdalhussein, E.; Jawad, R.K.; Madhloom AL-Salih, A.A.M.; Ahmed, M.A.; et al. Candle Flame Optimization: A Physics-Based Metaheuristic for Global Optimization. *Int. J. Intell. Eng. Syst.* **2025**, *18*, 826–837. [CrossRef]
3. Chao, G.; Sun, S.; Bi, J. A Survey on Multiview Clustering. *IEEE Trans. Artif. Intell.* **2021**, *2*, 146–168. [CrossRef]
4. Yang, Y.; Wang, H. Multi-view clustering: A survey. *Big Data Min. Anal.* **2018**, *1*, 83–107. [CrossRef]
5. Ng, A.; Jordan, M.; Weiss, Y. On spectral clustering: Analysis and an algorithm. *Adv. Neural Inf. Process. Syst.* **2001**, *14*, 849.
6. Nie, F.; Li, J.; Li, X. Parameter-free auto-weighted multiple graph learning: A framework for multiview clustering and semi-supervised classification. In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16), New York, NY, USA, 9–15 July 2016; Volume 9, pp. 1881–1887.
7. Wang, H.; Yang, Y.; Liu, B. GMC: Graph-Based Multi-View Clustering. *IEEE Trans. Knowl. Data Eng.* **2020**, *32*, 1116–1129. [CrossRef]
8. Liang, Y.; Huang, D.; Wang, C.D. Consistency meets inconsistency: A unified graph learning framework for multi-view clustering. In *2019 IEEE International Conference on Data Mining (ICDM)*; IEEE: New York, NY, USA, 2019; pp. 1204–1209.
9. Huang, S.; Tsang, I.W.; Xu, Z.; Lv, J. Measuring Diversity in Graph Learning: A Unified Framework for Structured Multi-View Clustering. *IEEE Trans. Knowl. Data Eng.* **2022**, *34*, 5869–5883. [CrossRef]
10. Li, Y.; Nie, F.; Huang, H.; Huang, J. Large-scale multi-view spectral clustering via bipartite graph. In Proceedings of the AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; Volume 29.
11. Kang, Z.; Zhou, W.; Zhao, Z.; Shao, J.; Han, M.; Xu, Z. Large-Scale Multi-View Subspace Clustering in Linear Time. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 4412–4419. [CrossRef]
12. Nie, F.; Wang, X.; Deng, C.; Huang, H. Learning a Structured Optimal Bipartite Graph for Co-Clustering. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 4132–4141.
13. Li, X.; Zhang, H.; Wang, R.; Nie, F. Multiview Clustering: A Scalable and Parameter-Free Bipartite Graph Fusion Method. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 330–344. [CrossRef]
14. Wang, S.; Liu, X.; Zhu, X.; Zhang, P.; Zhang, Y.; Gao, F.; Zhu, E. Fast Parameter-Free Multi-View Subspace Clustering with Consensus Anchor Guidance. *IEEE Trans. Image Process.* **2022**, *31*, 556–568. [CrossRef]
15. Kang, Z.; Lin, Z.; Zhu, X.; Xu, W. Structured Graph Learning for Scalable Subspace Clustering: From Single View to Multiview. *IEEE Trans. Cybern.* **2022**, *52*, 8976–8986. [CrossRef]
16. Fang, S.G.; Huang, D.; Cai, X.S.; Wang, C.D.; He, C.; Tang, Y. Efficient Multi-View Clustering via Unified and Discrete Bipartite Graph Learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *35*, 11436–11447. [CrossRef]
17. Liu, S.; Liao, Q.; Wang, S.; Liu, X.; Zhu, E. Robust and consistent anchor graph learning for multi-view clustering. *IEEE Trans. Knowl. Data Eng.* **2024**, *36*, 4207–4219. [CrossRef]
18. Yan, W.; Zhao, X.; Yue, G.; Ren, J.; Xu, J.; Liu, Z.; Tang, C. Diversity-induced bipartite graph fusion for multiview graph clustering. *IEEE Trans. Emerg. Top. Comput. Intell.* **2024**, *8*, 2592–2601. [CrossRef]
19. Li, S.; Liu, K.; Zheng, M.; Bai, L. Multi-view spectral clustering algorithm based on bipartite graph and multi-feature similarity fusion. *Neural Netw.* **2026**, *194*, 108177. [CrossRef] [PubMed]
20. Luo, M.; Nie, F.; Chang, X.; Yang, Y.; Hauptmann, A.G.; Zheng, Q. Discrete Multi-Graph Clustering. *IEEE Trans. Image Process.* **2019**, *28*, 4701–4712. [CrossRef] [PubMed]
21. Xia, W.; Gao, Q.; Wang, Q.; Gao, X.; Ding, C.; Tao, D. Tensorized bipartite graph learning for multi-view clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 5187–5202. [CrossRef]
22. Gu, W.; Guo, J.; Wang, H.; Zhang, G.; Zhang, B.; Chen, J.; Cai, H. Efficient multi-view clustering via essential tensorized bipartite graph learning. *IEEE Trans. Emerg. Top. Comput. Intell.* **2024**, *9*, 2952–2964. [CrossRef]
23. Long, Z.; Wang, Q.; Ren, Y.; Liu, Y.; Zhu, C. S2mvtc: A simple yet efficient scalable multi-view tensor clustering. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024; pp. 26213–26222.

24. Liu, J.; Liu, X.; Li, C.; Wan, X.; Tan, H.; Zhang, Y.; Liang, W.; Qu, Q.; Feng, Y.; Guan, R.; et al. Large-scale multi-view tensor clustering with implicit linear kernels. In Proceedings of the Computer Vision and Pattern Recognition Conference, Nashville, TN, USA, 11–15 June 2025; pp. 20727–20736.
25. Zhao, Z.; Wang, T.; Xin, H.; Wang, R.; Nie, F. Multi-view clustering via high-order bipartite graph fusion. *Inf. Fusion* **2025**, *113*, 102630. [CrossRef]
26. Jiang, H.; Tao, H.; Jiang, Z.; Hou, C. Unaligned multi-view clustering via diversified anchor graph fusion. *Pattern Recognit.* **2026**, *170*, 111977.
27. Ji, J.; Feng, S.; Li, Y. Tensorized unaligned multi-view clustering with multi-scale representation learning. In Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Barcelona, Spain, 25–29 August 2024; pp. 1246–1256.
28. Nie, F.; Wang, X.; Jordan, M.; Huang, H. The constrained laplacian rank algorithm for graph-based clustering. In Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016.
29. Huang, J.; Nie, F.; Huang, H. Spectral rotation versus k-means in spectral clustering. In Proceedings of the AAAI Conference on Artificial Intelligence, Bellevue, WA, USA, 14–18 July 2013; pp. 431–437.
30. von Luxburg, U. A tutorial on spectral clustering. *Stat. Comput.* **2007**, *17*, 395–416. [CrossRef]
31. Duchi, J.; Shalev-Shwartz, S.; Singer, Y.; Chandra, T. Efficient projections onto thel1-ball for learning in high dimensions. In *Proceedings of the 25th International Conference on Machine Learning—ICML '08*; ACM Press: New York, NY, USA, 2008; pp. 272–279. [CrossRef]
32. Fan, K. On a Theorem of Weyl Concerning Eigenvalues of Linear Transformations I. *Proc. Natl. Acad. Sci. USA* **1949**, *35*, 652–655. [CrossRef]
33. Mirsky, L. A trace inequality of John von Neumann. *Monatshefte Math.* **1975**, *79*, 303–306. [CrossRef]
34. Lee, Y.J.; Grauman, K. Foreground focus: Unsupervised learning from partially matching images. *Int. J. Comput. Vis.* **2009**, *85*, 143–166. [CrossRef]
35. Zhao, H.; Ding, Z.; Fu, Y. Multi-view clustering via deep matrix factorization. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; Volume 31.
36. Cao, X.; Zhang, C.; Fu, H.; Liu, S.; Zhang, H. Diversity-induced multi-view subspace clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 586–594.
37. Dua, D.; Graff, C. UCI Machine Learning Repository. University of California, Irvine, School of Information and Computer Sciences. 2019. Available online: http://archive.ics.uci.edu/ml (accessed on 10 May 2026).
38. Mallah, C.; Cope, J.; Orwell, J. Plant leaf classification using probabilistic integration of shape, texture and margin features. In *Proceedings of the 10th IASTED International Conference on Signal Processing, Pattern Recognition and Applications*; ACTA Press: Calgary, AB, Canada, 2013; pp. 45–54.
39. Nie, F.; Cai, G.; Li, X. Multi-view clustering and semi-supervised classification with adaptive neighbours. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; Volume 31.
40. Xu, W.; Liu, X.; Gong, Y. Document clustering based on non-negative matrix factorization. In Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval, Toronto, ON, Canada, 28 July–1 August 2003; pp. 267–273.
41. Strehl, A.; Ghosh, J. Cluster ensembles—A knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.* **2002**, *3*, 583–617.
42. Manning, C.D.; Raghavan, P.; Schütze, H. *Introduction to Information Retrieval*; Cambridge University Press: Cambridge, UK, 2008.
43. Kang, Z.; Shi, G.; Huang, S.; Chen, W.; Pu, X.; Zhou, J.T.; Xu, Z. Multi-graph fusion for multi-view spectral clustering. *Knowl.-Based Syst.* **2020**, *189*, 105102.

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.