A Comprehensive Survey on Spectral Clustering with Graph Structure Learning

Kamal Berahmand, Farid Saberi-Movahed, Razieh Sheikhpour, Yuefeng Li, and Mahdi Jalili, *Member, IEEE*

Abstract—**Spectral clustering is a powerful technique for clustering high-dimensional data, utilizing graph-based representations to detect complex, non-linear structures and non-convex clusters. The construction of a similarity graph is essential for ensuring accurate and effective clustering, making graph structure learning (GSL) central for enhancing spectral clustering performance in response to the growing demand for scalable solutions. Despite advancements in GSL, there is a lack of comprehensive surveys specifically addressing its role within spectral clustering. To bridge this gap, this survey presents a comprehensive review of spectral clustering methods, emphasizing on the critical role of GSL. We explore various graph construction techniques, including pairwise, anchor, and hypergraph-based methods, in both fixed and adaptive settings. Additionally, we categorize spectral clustering approaches into single-view and multi-view frameworks, examining their applications within one-step and two-step clustering processes. We also discuss multi-view information fusion techniques and their impact on clustering data. By addressing current challenges and proposing future research directions, this survey provides valuable insights for advancing spectral clustering methodologies and highlights the pivotal role of GSL in tackling large-scale and high-dimensional data clustering tasks.**

Index Terms—**Spectral Clustering, Graph Structure Learning, Spectral Embedding, Multi-view Clustering**

I. INTRODUCTION

CLUSTRING, is a fundamental technique in unsupervised learning, aimed at partitioning data points into distinct groups or clusters, such that points within a cluster are more similar to each other than to those in other clusters [1]–[3]. Unlike supervised learning, clustering operates without predefined labels or categories, instead identifying inherent patterns and structures within the data. This makes it particularly useful for exploratory data analysis, where the objective is to uncover hidden patterns without prior assumptions about the data structure [4]. Clustering is widely applied across various fields, including marketing [5], social network analysis [6], image segmentation [7], bioinformatics [8], anomaly detection [9], and document categorization [10]. It simplifies complex data,

Kamal Berahmand, Yuefeng Li are with School of Computer Sciences, Faculty of Science, Queensland University of Technology (QUT), Brisbane, Australia. (E-mail: kamal.berahmand@hdr.qut.edu.au, y2.li@qut.edu.au).

Farid Saberi-Movahed is with the Department of Applied Mathematics, Faculty of Sciences and Modern Technologies, Graduate University of Advanced Technology, Kerman, Iran (E-mail: f.saberimovahed@kgut.ac.ir).

Razieh Sheikhpour is with the Department of Computer Engineering, Faculty of Engineering, Ardakan University, P.O. Box 184, Ardakan, Iran (E-mail: rsheikhpour@ardakan.ac.ir).

Mahdi Jalili is with the School of Engineering, RMIT University, Melbourne, VIC 3000, Australia (E-mail: mahdi.jalili@rmit.edu.au).

Corresponding author: Razieh Sheikhpour.

enhances understanding, and often serves as a preprocessing step for other machine learning tasks, such as classification.

Clustering methods can be broadly classified into traditional and dimensionality reduction-based approaches as illustrated in Fig. 1. Traditional methods include partitioning-based [11], hierarchical [12], density-based [13], and probabilistic algorithms [14], each employing different strategies to group data. Partitioning-based methods, like K-means, divide data into a fixed number of clusters, each represented by a centroid [11]. Hierarchical methods, such as agglomerative and divisive clustering, build a cluster hierarchy by either merging smaller clusters (agglomerative) or splitting larger ones (divisive) [15], [16]. Density-based methods, like DBSCAN, group data points based on regions of high density, identifying clusters of arbitrary shapes [17]. Probabilistic methods, such as Gaussian Mixture Models (GMM), use probabilistic models to represent data distributions and clusters [14].
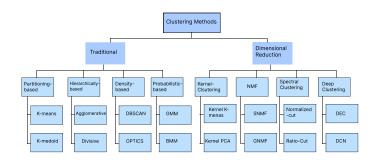

Fig. 1: Exploring Clustering Methods: From Traditional to Deep Learning Approaches.

While traditional methods are effective for lower-dimensional, well-structured datasets, they face limitations when applied to high-dimensional or complex data. In high-dimensional spaces, measuring distances between points becomes challenging, often resulting in poor clustering performance. Additionally, traditional methods often fail to capture non-convex shapes and intricate data structures. To address these limitations, dimensionality reduction-based clustering methods have emerged, reducing the number of features or dimensions and enabling clustering in a lower-dimensional space while preserving essential structural information.

Dimensionality reduction-based methods include Non-negative Matrix Factorization (NMF) [18], spectral clustering [19], [20], kernel clustering [21], and deep clustering [22]. NMF is an effective dimensionality reduction technique used for clustering that factorizes the data matrix into two lower-

dimensional non-negative matrices [18]. However, it may face challenges when dealing with more complex or non-linear data structures. Kernel clustering, including methods like Kernel K-means and Kernel PCA, handles non-linear relationships in the data by applying kernel functions [21]. Spectral clustering utilizes graph theory, representing data points as nodes and their similarities as edges, while leveraging methods such as Ratio-cut [23] and Normalized-cut [24]. Deep clustering integrates deep learning with clustering by using neural networks to learn lower-dimensional representations [22]. While deep clustering is powerful for large-scale, high-dimensional data, it requires significant computational resources and careful hyperparameter tuning.

Among dimensionality reduction techniques, spectral clustering stands out for its ability to handle complex data by identifying non-convex clusters and capturing non-linear structures through graph-based approaches. By representing data points as nodes in a graph and using graph-based embedding, spectral clustering partitions data based on connectivity and relationships. This flexibility makes spectral clustering applicable to a wide range of problems across various fields, particularly when combined with effective graph construction techniques. Spectral clustering is especially suitable for high-dimensional data, where spectral embedding reduces dimensionality while retaining essential structural information, mitigating challenges like the "curse of dimensionality" and enabling reliable clustering of non-linear patterns. For large-scale datasets, anchor graph-based spectral clustering offers a scalable solution by using a subset of representative points, or anchors, to approximate relationships efficiently, preserving computational resources while maintaining clustering quality. Consequently, spectral clustering is both versatile and scalable, adaptable to high-dimensional and large-volume data applications, making it a powerful tool for complex clustering tasks [21], [25].

A critical factor in the success of spectral clustering is the construction of the similarity graph, which serves as the foundation for the entire process. This graph represents relationships between data points, with nodes corresponding to the points and edges representing their pairwise similarities. The quality of this graph significantly influences the spectral embedding and, consequently, the clustering results, as it directly affects how accurately the underlying structure of the data is captured [25]. Common graph types used in spectral clustering include pairwise graphs [26], anchor graphs [27], [28], and hypergraphs [29], [30], each offering distinct advantages depending on the nature of the data. These graphs can either be fixed, where the structure remains constant throughout the clustering process, or adaptive, where the graph structure is dynamically learned and updated during the process.

Despite the advancements made in spectral clustering, particularly in fields like image segmentation [31], text classification [32], and industrial design [33], a comprehensive survey on spectral clustering that focuses on graph structure learning (GSL) is still lacking. To address this gap, our survey presents an extensive review of spectral clustering, with particular emphasis on graph structure and its critical role in improving clustering accuracy. While a previous survey [34] provides an overview of spectral clustering, focusing on graph cuts, Laplacian matrices, and the clustering process, our survey delves into the more specific and crucial aspect of GSL. Thye previous survey emphasizes the mathematical foundations and applications of spectral clustering but do not extensively explore the importance of how graphs are constructed and how this impacts clustering performance. In contrast, our survey highlights the role of graph construction techniques, including pairwise, anchor, and hypergraph methods, in both fixed and adaptive forms. Additionally, we classify spectral clustering methods into single-view and multi-view approaches, examining their applications within both one-step and two-step frameworks. The distinction between these frameworks lies in whether clustering is treated as an independent step following the spectral embedding or is jointly optimized with it. We also provide a deeper examination of information fusion in multi-view spectral clustering, an area not covered in previous survey, offering new insights for improving clustering in high-dimensional, complex data.

Our contributions in this survey are outlined as follows:

- For the first time, we present the most extensive and detailed survey on spectral clustering, with a particular emphasis on GSL, highlighting its importance for enhancing clustering accuracy.
- We provide a thorough review of various graph construction techniques, including pairwise, anchor, and hypergraphs, in both their fixed and adaptive forms. Additionally, we categorize spectral clustering methods into single-view and multi-view approaches, analyzing how different graph construction techniques are utilized in these approaches, and their application in both one-step and two-step clustering frameworks.
- We discuss information fusion techniques in multi-view spectral clustering, providing new insights into how integrating data from multiple sources can enhance clustering performance. This is particularly relevant for handling complex, heterogeneous, and high-dimensional data, making it an important contribution to advancing multi-view spectral clustering.
- We present an analysis of recent research advancements, emerging challenges, and potential future directions in spectral clustering and graph learning. This includes identifying promising areas for further research and the development of more advanced algorithms, contributing to the growth and evolution of the field.

The remainder of this paper is structured as follows: In Section II, we provide a detailed background on spectral clustering. Section III presents the main stages of spectral clustering, including graph structure learning, spectral embedding, and the partitioning phase. Section IV offers a comprehensive categorization of spectral clustering methods based on single-view and multi-view approaches, graph structure type, and one-step versus two-step frameworks, and explores different spectral clustering methods within each category. Section V discusses future research directions, and finally, Section VI presents the conclusion.

II. Background of Spectral Clustering

A. Notations and Preliminaries

Table I provides a summary of the essential notations and initial concepts utilized throughout this paper. The data matrix, defined as $X = [x_1, x_2, \ldots, x_n]$, contains n samples, each denoted by $x_i \in \mathbb{R}^{d \times 1}$, where $i = 1, \ldots, n$. In the case of multi-view scenarios, the dataset is considered as $\mathcal{D} = \{X^1, X^2, \ldots, X^V\}$, which includes V views. The data in the v-th view is represented by $X^v \in \mathbb{R}^{d_v \times n}$, with n as the sample count and d_v as the number of features in the v-th view, for $v = 1, \ldots, V$.

TABLE I: Summary of Key Notations

Notation	Description
$X = \{x_1, \ldots, x_n\}$	Input data set with n samples
$x_i \in \mathbb{R}^{d \times 1}$	The i-th sample, for $i = 1, \ldots, n$
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	Graph with the set of vertices $\mathcal{V}$ and the set of edges $\mathcal{E}$
F	Cluster indicator matrix
$\mathcal{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_k\}$	The set of clusters
$vol(\mathcal{C}_l)$	Volume of a cluster $\mathcal{C}_l$, for $l = 1, \ldots, k$
k	Number of clusters
L	Graph Laplacian matrix
L_{sym}	Symmetric graph Laplacian matrix
L_{rw}	Random Walk graph Laplacian matrix
L_h	Normalized hypergraph Laplacian matrix
W	Weighted adjacency matrix of $\mathcal{G}$
D	Degree matrix of $\mathcal{G}$
A	Similarity matrix
S	Similarity matrix (for adaptive methods)
P	Projection matrix
Z	Self-expressive matrix
I_p	Identity matrix of size p
$\|X\|_F$	The Frobenius norm, $\|X\|_F = \sqrt{\sum_{i=1}^d \sum_{j=1}^n x_{ij}^2}$
$\|X\|_1$	The ℓ_1 norm, $\|X\|_1 = \sum_{i=1}^d \sum_{j=1}^n |x_{ij}|$
$\|X\|_*$	The Nuclear norm of X
$\|X\|_{21}$	The ℓ_{21} norm, $\|X\|_{21} = \sum_{i=1}^d \sqrt{\sum_{j=1}^n x_{ij}^2}$
$\|x_i\|_2$	The Euclidean norm of x_i
X^T	The transpose operator of X
$Tr(\cdot)$	The trace operator
Ind	The set of indicator matrices

B. Graph-cut

Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ represents the finite set of vertices and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes the set of edges, indicating the pairwise connections among vertices. Moreover, suppose that $W \in \mathbb{R}^{n \times n}$ stands for the weighted adjacency matrix of $\mathcal{G}$ and D for its degree matrix. The goal of a graph-cut algorithm is to identify a partition $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k\}$ of the vertex set $\mathcal{V}$ such that $\bigcup_{i=1}^k \mathcal{C}_i = \mathcal{V}$, ensuring that the union of all clusters encompasses the entire vertex set [19]. Furthermore, $\mathcal{C}_i \cap \mathcal{C}_j = \emptyset$ for all $i \neq j$, which guarantees that the clusters are disjoint, so each vertex belongs to only one cluster. Two effective formulations of the graph-cut problem are Ratio-cut (R-cut) [23] and Normalized-cut (N-cut) [24], which differ in how they handle cluster balance. For further explanation:

(i) **R-cut:** The R-cut approach divides the graph into the k clusters $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k$ by minimizing the cut value for each cluster, normalized by the number of nodes within it. This can be expressed as [23]:

$$\min_{\mathcal{C}_1, \ldots, \mathcal{C}_k} Rcut(\mathcal{C}_1, \ldots, \mathcal{C}_k) = \sum_{l=1}^k \frac{cut(\mathcal{C}_l, \bar{\mathcal{C}}_l)}{|\mathcal{C}_l|}, \quad (1)$$

where $cut(\mathcal{C}_l, \bar{\mathcal{C}}_l)$ indicates the total edge weight connecting the nodes in the cluster $\mathcal{C}_l$ to those outside it, and $|\mathcal{C}_l|$ is the number of nodes in $\mathcal{C}_l$.

Let us also assume that $L = D - W$ represents the graph Laplacian matrix, and that f_l denotes the cluster indicator vectors with n elements, defined as $f_{il} = \frac{1}{\sqrt{|\mathcal{C}_l|}}$ if $v_i \in \mathcal{C}_l$; otherwise, $f_{il} = 0$, for $i = 1, \ldots, n$ and $l = 1, \ldots, k$. Accordingly, the objective function of the R-cut problem can be restated as:

$$Rcut(\mathcal{C}_1, \ldots, \mathcal{C}_k) = \sum_{l=1}^k \frac{f_l^T L f_l}{f_l^T f_l}. \quad (2)$$

With this description, the R-cut problem is formulated as:

$$\min_{\mathcal{C}_1, \ldots, \mathcal{C}_k} Tr(F^T L F), \quad (3)$$

where $F = [f_1, \ldots, f_k]$ is the cluster indicator matrix and $F^T F = I_k$.

(ii) **N-cut:** In contrast, the N-cut method adjusts each cut value according to the overall connectivity of the cluster, defined as the cluster volume, represented by $vol(\mathcal{C}_l)$, where $vol(\mathcal{C}_l) = \sum_{v_i \in \mathcal{C}_l} d_i$, and d_i is the degree of the i-th node. The N-cut problem is formulated as [24]:

$$\min_{\mathcal{C}_1, \ldots, \mathcal{C}_k} Ncut(\mathcal{C}_1, \ldots, \mathcal{C}_k) = \sum_{l=1}^k \frac{cut(\mathcal{C}_l, \bar{\mathcal{C}}_l)}{vol(\mathcal{C}_l)}. \quad (4)$$

By minimizing $Ncut$, this approach promotes clusters that are densely connected internally while being loosely connected to other clusters. Similar to R-cut, the N-cut problem is also formulated using the Laplacian matrix as:

$$\min_{\mathcal{C}_1, \ldots, \mathcal{C}_k} Tr(F^T L F), \quad (5)$$

where $F = [f_1, \ldots, f_k]$ represents the cluster indicator matrix. The components of f_l are defined such that $f_{il} = \frac{1}{\sqrt{vol(\mathcal{C}_l)}}$ if $v_i \in \mathcal{C}_l$; otherwise, $f_{il} = 0$ for $i = 1, \ldots, n$ and $l = 1, \ldots, k$. Furthermore, it follows that $F^T D F = I_k$.

Although both the R-cut and N-cut are fundamental in graph clustering, their inherent NP-hardness arises from the discrete nature of the cluster indicator matrix F [19]. To address this challenge, a relaxation technique can be employed that allows F to assume real values, transforming the discrete optimization objectives into continuous forms that can be efficiently solved using the eigenvalue decomposition. Specifically, the continuous relaxations can be presented through the following formulations:

- **R-cut relaxation:**

$$\min_{F \in \mathbb{R}^{n \times k}, \, F^T F = I_k} Tr(F^T L F). \quad (6)$$

- **N-cut relaxation:**

$$\min_{F \in \mathbb{R}^{n \times k}, \, F^T D F = I_k} Tr(F^T L F). \quad (7)$$

III. Main Stages of Spectral Clustering

Spectral clustering has emerged as a sophisticated graph-cut methodology in machine learning, distinguished by its ability to capture intrinsic manifold structures through the eigenspectrum analysis of graph Laplacian matrices [1]. The mathematical foundation of spectral clustering integrates graph theory and manifold learning principles, enabling effective

handling of complex, nonlinear data distributions while maintaining theoretical guarantees and computational efficiency in the clustering process. As illustrated in Fig. 2, spectral clustering transcends traditional clustering methods by establishing the clustering objective into three theoretically-grounded stages: graph structure learning for optimal relationship modeling [34], spectral embedding through eigendecomposition for manifold structure revelation [35], and partitioning process for final cluster assignment [36]. Each of these stages will be discussed in detail in the following sections.
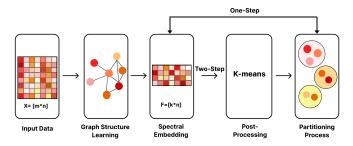

Fig. 2: Systematic Overview of Spectral Clustering Framework.

A. Graph Structure Learning

Graph structure learning (GSL) constitutes a fundamental paradigm in machine learning, focusing on the optimization of graph representations from empirical data [37]. As an essential component across multiple domains including Graph Neural Networks (GNNs) [38], manifold learning [39], and graph-based semi-supervised learning [40], GSL stands as a cornerstone methodology for representing complex data relationships. Within the spectral clustering framework, GSL serves as a crucial component, where the quality of the learned graph structure directly determines the efficacy of subsequent spectral embedding and partitioning phase.

The taxonomy of GSL methodologies in spectral clustering encompasses two paradigms, fixed and adaptive, each offering distinct approaches to graph construction. Fixed methodologies employ predetermined protocols and parameters for graph construction, which enhances computational efficiency and generalization across datasets. In contrast, adaptive methodologies dynamically optimize the graph structure by designing a tailored optimization problem, enabling them to capture dataset-specific characteristics and provide greater representational flexibility for complex topological relationships. Selecting an appropriate methodology has a significant impact on clustering performance, as the resulting graph structure supports spectral embedding and subsequent partition optimization. As illustrated in Fig. 3, GSL can be systematically categorized into three principal architectures: pairwise graphs, anchor-based graphs, and hypergraphs, each of which includes both fixed and adaptive methodological variants.

1) Pairwise Graph: The foremost and undeniably crucial step in spectral clustering involves constructing a similarity matrix. This matrix must reflect the pairwise similarities
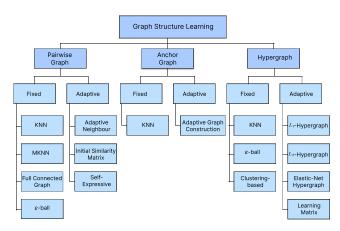

Fig. 3: Graph Structure Learning Methods.

among the data points, meaning that points within the same cluster should exhibit strong connections with high similarity values, unlike points belonging to different clusters. When constructing pairwise graphs from a dataset with the pairwise similarities, the primary objective is to capture either the local neighborhood relationships between data points or the broader, global structure of the dataset. The methodologies for constructing pairwise graphs can be systematically categorized into two approaches: fixed [41] and adaptive [42], each of which is explained below.

a) Fixed Methods: Suppose that the set of data points to be partitioned into k subsets is indicated by $\{x_1, \ldots, x_n\}$. To facilitate spectral clustering, the data must first be represented in the form of a similarity graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. In this graph, each vertex v_i is associated with a corresponding data point x_i, for $i = 1, \ldots, n$, while $\mathcal{E}$ represents the edges that connect the vertices. Furthermore, an $n \times n$ similarity matrix $A = [a_{ij}]$ is established, in which the values of a_{ij} represent the pairwise similarities between the data points x_i and x_j. In pairwise graph construction, fixed methodologies refer to the approaches in which the graph structure is determined by fixed criteria instead of being dynamically adjusted to the data [41]. These approaches provide computational efficiency while maintaining consistent structural properties across different datasets. The construction of a fixed-based pairwise graph is systematically performed through four primary principles [19]:

(i) **K-Nearest Neighbors (KNN):** It links each data point to its K nearest neighbors based on a specified distance metric,such as the Euclidean distance. Consequently, a graph is created where the edges denote the closest K points to each node.

(ii) **Mutual KNN (MKNN):** It builds upon KNN by creating an edge between two points only if each point is included in the KNN of the other. This guarantees a reciprocal connection, thus strengthening the relationships represented in the graph.

(iii) **Fully Connected:** It describes a complete graph in which every distinct pair of nodes is interconnected by an edge, leading to a dense network of links.

(iv) ε**-Ball:** This refers to a spherical area surrounding a point

in a metric space, including the Euclidean metric space, characterized by a radius ε. It encompasses all points that are within a distance of ε from the specified point, forming edges between the central point and all points contained in the ε-ball.

Each of the methods described above constructs the similarity matrix $\mathbf{A} = [a_{ij}]$ based on different criteria, including [19]:

(i) **Binary Similarity:** In this approach, the pairwise similarity a_{ij} is defined as $a_{ij} = 1$ if the data points x_i and x_j are linked to each other, while $a_{ij} = 0$ otherwise.

(ii) **Cosine Similarity:** This criterion evaluates the cosine of the angle formed by two data points. It is computed as $a_{ij} = \frac{x_i^T x_j}{\|x_i\|_2 \|x_j\|_2}$.

(iii) **Gaussian Kernel:** This technique establishes pairwise similarity through the Gaussian kernel function, calculated as $a_{ij} = \exp\left(-\frac{\|x_i - x_j\|_2^2}{2\sigma^2}\right)$, where σ is the bandwidth parameter that regulates the width of the Gaussian function.

b) Adaptive Methods: Adaptive methods dynamically learn the graph structure based on data characteristics during the clustering process [43]. Unlike fixed methods, these approaches optimize the graph structure to better reflect intrinsic data relationships and adapt to local data distributions. This flexibility enables more accurate representation of complex data structures. The main adaptive approaches include:

Adaptive Neighbor. Compared to KNN-based graph construction, the adaptive neighbor approach provides a significant advancement by simultaneously learning both the neighbors of each data point and their respective weights from the original data. This adaptive mechanism selects optimal neighbors to ensure that points in closer proximity have stronger relationships. A particularly promising model in this context is the Clustering with Adaptive Neighbors (CAN) method [42], which formalizes this adaptive process within a probabilistic optimization framework, introduced as:

$$\min_{s_{ij}} \sum_{i,j=1}^{n} \left(\|x_i - x_j\|_2^2 s_{ij} + \gamma s_{ij}^2\right) \tag{8}$$

$$\text{s.t.} \quad \sum_{j=1}^{n} s_{ij} = 1, 0 \leq s_{ij} \leq 1,$$

where γ is the regularization parameter, and the matrix $S = [s_{ij}] \in \mathbb{R}^{n \times n}$ represents neighbor assignments, with each s_{ij} indicating the probability that the data sample x_j is a neighbor of x_i. The effectiveness of the CAN framework is further improved by incorporating a projection matrix $P \in \mathbb{R}^{d \times m}$, where $m \leq d$, to tackle the challenges associated with clustering high-dimensional data. This novel approach, referred to as Projected CAN (PCAN) [42], seeks to optimize both dimensionality reduction and neighborhood learning simultaneously through the following problem:

$$\min_{s_{ij}, P} \sum_{i,j=1}^{n} \left(\|P^T x_i - P^T x_j\|_2^2 s_{ij} + \gamma s_{ij}^2\right) \tag{9}$$

$$\text{s.t.} \quad P \in \mathbb{R}^{d \times m}, \sum_{j=1}^{n} s_{ij} = 1, 0 \leq s_{ij} \leq 1, P^T B P = I_m,$$

where B represents the total scatter matrix. Additionally, the imposition of the orthogonality constraint $P^T B P = I_m$ guarantees statistical uncorrelatedness of the data in the subspace.

Initial Similarity Matrix. This approach iteratively refines an initial similarity matrix $A \in \mathbb{R}^{n \times n}$ to achieve more accurate similarity relationships [44]. It specifically employs two distinct distance constraints–the Frobenius norm and the L_1 norm–to enhance the initial similarity matrix A into a refined similarity matrix $S \in \mathbb{R}^{n \times n}$. Consequently, this learning framework is expressed as follows:

- **The Frobenius norm-based problem:**

$$\min_{S} \|S - A\|_F^2 \tag{10}$$

$$\text{s.t.} \quad S = [s_{ij}] \in \mathbb{R}^{n \times n}, \sum_{j=1}^{n} s_{ij} = 1, 0 \leq s_{ij} \leq 1.$$

- **The L_1 norm-based problem:**

$$\min_{S} \|S - A\|_1 \tag{11}$$

$$\text{s.t.} \quad S = [s_{ij}] \in \mathbb{R}^{n \times n}, \sum_{j=1}^{n} s_{ij} = 1, 0 \leq s_{ij} \leq 1.$$

Here, the constraints $\sum_{j=1}^{n} s_{ij} = 1$ and $0 \leq s_{ij} \leq 1$ guarantee proper probability distributions. Moreover, the choice of different norm minimizations introduces versatility in adjusting the refinement approach: the Frobenius norm focuses on reducing the Euclidean distance between S and A, leading to smoother updates in similarity, whereas the L_1 norm promotes sparsity, which can produce a clearer and more sparse similarity structure in S.

Self-expressive. Self-expressive learning is an advanced approach for identifying intrinsic relationships between data points, especially useful when the data is situated across several subspaces [45]. This technique creates a self-expressive matrix Z, which illustrates how each data point can be optimally reconstructed using the other points in the dataset, thereby uncovering the underlying subspace structures through a global representation. The self-expressive framework is fundamentally expressed as:

$$\min_{Z} \|X - f(X)Z\|_l + \Omega(X, Z) \tag{12}$$

$$\text{s.t.} \quad Z \in \mathbb{R}^{n \times n}, Z \in \mathcal{C},$$

where $\|\cdot\|_l$ denotes a specified matrix norm, $f(X)$ represents a matrix function (typically defined as $f(X) = X$), $\Omega(X, Z)$ is a regularization term, and $\mathcal{C}$ defines the constraint set. A key outcome of Problem (12) is the formulation of a similarity matrix S, expressed as $S = (|Z| + |Z^T|)/2$. This formulation aids in uncovering intrinsic subspace structures via the reconstruction term and different regularization methods. Furthermore, the symmetrization of Z ensures a well-defined similarity measure, which can be valuable for effective clustering applications.

2) Anchor Graph: Anchor graphs present a way to construct a graph by selecting a smaller, representative subset of nodes, known as "anchors," rather than incorporating every node from a large dataset [46]. These anchors facilitate

efficient information propagation across the graph. Centering on these anchors enables the graph to keep core relationships and structures intact without processing all nodes, improving computational speed and making analysis more manageable. The construction of an anchor graph framework involves three main stages: anchor point selection, anchor graph construction, and similarity matrix generation.

a) Anchor Point Selection: Various strategies can be employed to select a set of m anchor points from a dataset containing n samples, where $m \ll n$. Some of the most common examples include [46], [47]:

(i) **Random Selection:** It selects a set of m anchor points arbitrarily, offering simplicity and low computational cost, but may produce unrepresentative anchors, leading to weaker clustering results.

(ii) **K-means (KM):** It selects anchor points by clustering the data into m groups and using the centroids as anchors. This method improves anchor representativeness by focusing on denser data regions but requires multiple iterations and is sensitive to initial cluster centers.

(iii) **K-means++ (KM++):** It refines KM initialization by randomly selecting the first anchor and choosing subsequent points based on their squared distance from the nearest selected anchor, enhancing diversity and improving clustering performance.

(iv) **Balanced K-Means via Hierarchical K-Means (BKHK):** The BKHK algorithm creates representative anchors for large datasets by employing a balanced binary tree structure. It repeatedly divides the data into two equal clusters using a binary k-means approach, which maintains equal cluster sizes. The algorithm minimizes a distance-based formulation related to cluster centers, assigning samples to clusters according to the smallest distance differences. This hierarchical balanced method is a key feature of the BKHK algorithm.

b) Anchor Graph Construction: Anchor-based graph construction is based on two principal approaches [48]: fixed methods with preset rules and adaptive techniques that dynamically determine connection strengths.

Fixed Construction. The fixed construction of an anchor graph is a structured methodology that involves two sequential steps: neighborhood identification and edge weight assignment. Given data points $\{x_1, \ldots, x_n\}$ and anchor points $\{u_1, \ldots, u_m\}$, the process is detailed as follows [48]:

(i) **Neighborhood Identification:** Each data point x_i is connected to its r nearest anchor points through KNN, establishing the fundamental topology of the anchor graph.

(ii) **Edge Weight Assignment:** The weight measure w_{ij} between each data point x_i and anchor point u_j, where $i = 1, \ldots, n$ and $j = 1, \ldots, m$, is calculated based on one of the following three main schemes [48]:

- **Binary Weight:** Assigns $w_{ij} = 1$ if u_j is among the r nearest anchors of x_i, and $w_{ij} = 0$ otherwise.
- **Polynomial Kernel:** Defines w_{ij} as $w_{ij} = ((x_i)^T u_j)^d$ when u_j is one of the r nearest anchors to x_i, and $w_{ij} = 0$ otherwise. Here, the variable d indicates the degree of the polynomial used in the kernel function.

- **Gaussian Kernel:** Computes w_{ij} as $w_{ij} = \exp(-\frac{\|x_i - u_j\|_2^2}{2\sigma^2})$ if u_j is among the nearest r anchors of x_i, setting $w_{ij} = 0$ otherwise.

Adaptive Construction. In anchor graph construction, adaptive approaches extend the principles of adaptive neighbor methods used in pairwise graphs. The main objective is to learn the neighbors and weights of each data sample based on its relationships with anchor points, aiming to enhance the representation of both local and global data structures. The optimization problem for adaptive anchor graph learning on each data sample x_i (for $i = 1, \ldots, n$) is defined as [48]:

$$\min_{w_{ij}} \sum_{j=1}^{m} (\|x_i - u_j\|_2^2 w_{ij} + \gamma w_{ij}^2) \qquad (13)$$
$$\text{s.t.} \quad \sum_{j=1}^{n} w_{ij} = 1, \, w_{ij} > 0,$$

where w_{ij} encodes the learned weight between the data sample x_i and the anchor point u_j, and γ is a regularization parameter.

c) Similarity Matrix Generation: The generation of similarity matrices from anchor-based representations is a crucial step in establishing effective relationships between the data and anchor points. Two primary approaches have emerged for computing the similarity matrix within an anchor graph framework: the Gram matrix formulation [49] and the bipartite graph construction [46]. Each of them demonstrates theoretical elegance in representing complex data relationships while maintaining computational efficiency. For additional details,

(i) **Gram Matrix Formulation:** The inner product is a key concept in linear algebra for assessing the similarity between two entities. When two entities, such as vectors, have a high inner product, it means they point in a similar direction or share similar features. A practical application of inner products in representation learning involves constructing Gram matrices, which contain the inner products between all pairs of data points. This Gram matrix can be regarded as a similarity matrix, indicating how each data point relates to others. Let $W = [w_{ij}] \in \mathbb{R}^{n \times m}$ be a weight matrix constructed between the data and anchors points, where each w_{ij} indicates the similarity between the i-th data sample and the j-th anchor point, for $i = 1, \ldots, n$ and $j = 1, \ldots, m$. Then the associated Gram matrix $S \in \mathbb{R}^{n \times n}$, encoding the pairwise similarities induced by the anchor-based representation, is established as [49]:

$$S = WW^T. \qquad (14)$$

(ii) **Bipartite Graph Construction:** Constructing a bipartite graph serves as an effective method for modeling the relationships between samples and anchors [50]. To illustrate this, let $\mathcal{G}$ be a bipartite graph consisting of $n + m$ nodes, with the left-side nodes corresponding to the n data samples and the right-side nodes representing the m anchor points. Additionally, assume that $W = [w_{ij}] \in \mathbb{R}^{n \times m}$ denotes the weight matrix for $\mathcal{G}$. Based on this framework,

the associated similarity matrix $S \in \mathbb{R}^{(n+m)\times(n+m)}$ is defined as [51]:

$$S = \begin{bmatrix} 0 & W \\ W^T & 0 \end{bmatrix}. \tag{15}$$

3) Hypergraph: Assume that $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ presents a weighted hypergraph, where $\mathcal{V}$ is the vertex set, $\mathcal{E}$ is the set of hyperedges, and w is the weight vector for the hyperedges, where each element represents the positive weight assigned to each hyperedge [52]. The degree of each vertex $v \in \mathcal{V}$ and the degree of each hyperedge $e \in \mathcal{E}$ are respectively defined as $d(v) = \sum_{e \in \mathcal{E}} w(e)\, h(v,e)$ and $d(e) = \sum_{v \in \mathcal{V}} h(v,e)$, where $w(e)$ represents the weight assigned to each hyperedge e, and $h(v,e)$ is an element of the incidence matrix $H \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{E}|}$, which describes the connections between vertices and hyperedges. The entries of H are specified by $h(v,e) = 1$ if $v \in e$; otherwise, $h(v,e) = 0$. Let $W \in \mathbb{R}^{|\mathcal{E}| \times |\mathcal{E}|}$ be the diagonal matrix formed from the weight vector w for the hyperedges of G. Let us also define $D_v \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ and $D_e \in \mathbb{R}^{|\mathcal{E}| \times |\mathcal{E}|}$ as the diagonal matrices constructed from the vertex degrees $d(v)$ and the hyperedge degrees $d(e)$, respectively. The normalized Laplacian matrix for the hypergraph is expressed as $L = I - D_v^{\frac{-1}{2}} HWD_e^{-1}H^T D_v^{\frac{-1}{2}}$ [52].

a) Hypergraph Learning: Hypergraphs are effective for representing complex associations by linking multiple nodes simultaneously, thereby capturing higher-order relationships among vertices [53]. Hypergraph learning encompasses the construction and analysis of hypergraphs to represent intricate relationships within data. An essential component of hypergraph learning is the construction of the incidence matrix, which illustrates the connections between vertices and hyperedges [54]. By learning the incidence matrix, vertex-to-hyperedge connections and hyperedge strengths can be optimized, providing adaptable frameworks for hypergraph construction that capture complex, high-order data relationships. The methods for generating the incidence matrix can be generally divided into two primary categories [55]: fixed and adaptive methods. Fixed methods usually establish the incidence matrix using predetermined criteria or algorithms, and adaptive methods focus on deriving the optimal structure of the hypergraph directly from the data. Each of these categories will be discussed below.

(i) **Fixed Methods [55]:** Fixed methods for generating incidence matrices in hypergraph learning utilize predetermined algorithms to establish connections between vertices and hyperedges. There are three primary strategies employed in this context:

- **Nearest-neighbor-based Strategy [56]:** It calculates the distances between vertices and uses either the KNN or ε-ball criteria to identify neighbors. Hyperedges are then formed as follows $e(v) = \mathcal{N}(v) \cup \{v\}$, where $\mathcal{N}(v)$ represents the neighbors of the vertex v. Consequently, the hyperedge set is defined as $\mathcal{E} = \{e(v) \mid v \in \mathcal{V}\}$.
- **Clustering-based Strategy [57]:** It involves grouping vertices using clustering algorithms, such as k-means. For k clusters denoted as $\mathcal{V}_1, \ldots, \mathcal{V}_k$, hyperedges are constructed as $e_l = \mathcal{V}_l$ where $l = 1, \ldots, k$, resulting in the hyperedge set $\mathcal{E} = \{e_1, e_2, \ldots, e_k\}$.

(ii) **Adaptive Methods [55]:** Adaptive methods aim to learn key elements of a hypergraph, such as the weight matrix, vertex degree matrix, hyperedge degree matrix, and incidence matrix. These approaches are particularly effective in capturing the complex relationships between vertices and hyperedges, enabling accurate representation of intricate data structures. Among various adaptive methods for learning the incidence matrix, a notable variant is the self-expressive-based method, which typically operates in a two-step process [53]. First, it addresses a self-expressive problem; subsequently, it constructs the incidence relationship between hyperedges and their corresponding vertices using the representation coefficient matrix derived from this self-expressive problem.

To illustrate the function of self-expressive-based methods, let us first consider that each data sample x_i, for $i = 1, \ldots, n$, represents a vertex within the hypergraph. Additionally, it is assumed that the number of nodes equals the number of hyperedges, meaning $|\mathcal{V}| = |\mathcal{E}| = n$. The three primary variants of self-expressive-based methods can be categorized as follows:

- ℓ_1-**Hypergraph (ℓ_1-HG) [58]:** It integrates a self-expressive problem using the KNN information of each vertex v within the hypergraph and applies the ℓ_1-norm to learn a sparse representation coefficient matrix for constructing the hypergraph. The ℓ_1-HG framework is formulated as follows:

$$\min_z \|X(v) - Mz\|_2^2 + \alpha\|z\|_1 \tag{16}$$
$$\text{s.t.} \quad z \geq 0,$$

where z is the coefficient vector learned for the i-th nearest neighbor, and α is the regularization parameter. Moreover, $X(v)$ denotes the feature vector of the vertex v within the hypergraph, and M represents its associated KNN.

- ℓ_2-**Hypergraph (ℓ_2-HG) [59]:** It formulates the self-expressive problem by incorporating a locality-preserving constraint on the representation coefficient matrix, enabling the capture of high-order correlations within the data. The ℓ_2-HG framework is

$$\min_Z \|X - XZ - E\|_F^2 + \alpha\|Z\|_F^2 + \beta\|Q \odot Z\|_F^2 + \gamma\|E\|_1$$
$$\text{s.t.} \quad Z \in \mathbb{R}^{n \times n}, Z^T \mathbf{1}_n = \mathbf{1}_n, \quad \text{diag}(Z) = 0, \tag{17}$$

where E represents the matrix of data errors, and Q retains the local manifold structures. In (17), $\odot$ denotes the Hadamard product, $\mathbf{1}_n$ is an all-ones vector of size n, and α, β, γ are the parameters for regularization.

- **Elastic-Net Hypergraph (EN-HG) [60]:** It utilizes a combination of the ℓ_1 norm and the Frobenius norm to learn a sparse representation coefficient matrix. The EN-HG framework is formulated as follows:

$$\min_Z \|X - XZ\|_{21} + \|Z\|_1 + \|Z\|_F^2 \tag{18}$$
$$\text{s.t.} \quad Z \in \mathbb{R}^{n \times n}, \text{diag}(Z) = 0.$$

Finally, self-expressive-based methods construct the incidence relationship between hyperedges and their corresponding vertices using the representation coefficient matrix $Z =$

$[z_{ij}]$ derived from this self-expressive problem. In this case, two strategies are employed as follows [59]:

- **(0-1) Incidence Relationship:** This strategy is commonly utilized, where coefficients exceeding a specified threshold are retained. The incidence relation is defined as:

$$h(v_i, e_j) = \begin{cases} 1, & \text{if } |z_{ij}| \geq \theta_1 \\ 0, & \text{otherwise} \end{cases}, \qquad (19)$$

where $i, j = 1, \ldots, n$. In this context, the data sample x_i is linked to the hyperedge e_j based on the condition that z_{ij} surpasses the threshold θ_1.

- **Probability Incidence Relationship:** In this strategy, the incidence relation is defined through the representation coefficients as follows:

$$h(v_i, e_j) = \begin{cases} |z_{ij}|, & \text{if } |z_{ij}| \geq \theta_2 \\ 0, & \text{otherwise} \end{cases}, \qquad (20)$$

where θ_2 is a specified threshold.

b) Matrix Learning Techniques: Matrix learning techniques focus on enhancing the hypergraph structural representations through the formulation and solution of an optimization problem. The general framework for a matrix learning technique in hypergraph learning can be outlined as follows [61]:

$$\min_{H, D_e, D_v, W} f(X, H, D_e, D_v, W) + \Omega(W) \qquad (21)$$
$$\text{s.t.} \quad W \in \mathcal{C},$$

where f denotes a specific function related to (X, H, D_e, D_v, W), while $\Omega(W)$ and $\mathcal{C}$ denote a regularization term and a set of constraints for W, respectively.

In matrix learning techniques, a common approach for constructing hyperedges is the threshold-based strategy, which allows each sample to have a variable number of nearest neighbors [61]. Using this approach, the hyperedge $e(v_j)$ is defined as follows:

$$e(v_j) = \{v_i \in \mathcal{V} \mid \theta(g(x_i), g(x_j)) \leq \sigma_j\}, \qquad (22)$$

where $i, j = 1, \ldots, n$, and σ_j is a predefined threshold. Here, $\theta(g(x_i), g(x_j))$ denotes the similarity measure between $g(x_i)$ and $g(x_j)$, where g is a defined mapping applied to the data samples. The resulting hyperedges are then utilized to build the incidence matrix.

B. Spectral Embedding

Spectral embedding is a prominent approach within dimensionality reduction techniques, rooted in the principles of spectral graph theory and manifold learning to uncover fundamental data structures [62]. By leveraging the eigenspectrum of graph Laplacian matrices, this method effectively captures complex nonlinear relationships and maintains local geometric characteristics, which are often missed by traditional linear methods. The effectiveness of spectral embedding lies in its ability to reveal the hidden manifold structure through graph spectral analysis, facilitating dimensionality reduction while retaining distinctive features and preserving topological connections within the embedded space [34].

Given a similarity graph that encodes the pairwise relationships between data points $\{x_1, \ldots, x_n\}$, the objective for spectral embedding optimization can be formulated as:

$$\min_{F} Tr(F^T L F) \qquad (23)$$
$$\text{s.t.} \quad F \in \mathbb{R}^{n \times k}, F^T F = I_k,$$

where F denotes the spectral embedding matrix, and k represents the number of dimensions into which the original data is embedded. Moreover, the graph Laplacian matrix L can be constructed in three principal forms [19]:

- **Unnormalized Graph Laplacian Matrix:**

$$L = D - W.$$

- **Normalized Graph Laplacian Matrix (Symmetric):**

$$L_{sym} = I_n - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}.$$

- **Normalized Graph Laplacian Matrix (Random Walk):**

$$L_{rw} = I_n - D^{-1} W.$$

It should be highlighted that the spectrum of the graph Laplacian matrix L reveals information about the connectivity of the graph. The orthogonality condition $F^T F = I_k$ in spectral embedding ensures that the embedding space is constructed from the eigenvectors corresponding to the smallest non-zero eigenvalues of L. Selecting these eigenvectors not only preserves the connectivity within clusters but also provides a low-dimensional representation that reflects the cluster structure of the graph.

C. Partitioning Process

In spectral clustering, the partitioning process assigns cluster memberships by leveraging a low-dimensional spectral representation obtained through spectral embedding. Partitioning approaches are classified into two main types–two-step and one-step clustering–depending on how spectral embedding and clustering are conducted [63].

1) Two-Step Clustering: In conventional spectral clustering approaches, the clustering task is generally conducted in two primary steps. Initially, these methods relax the discrete constraints and apply eigenvalue decomposition to the graph Laplacian matrix, resulting in a spectral embedding matrix F in a continuous form. Next, a clustering method such as K-means is performed on F to transform these continuous embeddings into discrete cluster assignments.

While two-step clustering presents a straightforward framework, the resulting optimal solution F may not closely resemble the true discrete clustering solution, as spectral embeddings do not inherently exhibit the required discrete structure. Additionally, K-means is sensitive to its initial settings and often converges to locally optimal solutions. Thus, using the obtained embedding matrix F as the basis for clustering may introduce inaccuracies, potentially compromising the clustering results.

2) One-Step Clustering: One-step spectral clustering algorithms utilize a unified framework that combines the learning of spectral embeddings with cluster assignments into a single process. This simultaneous process allows for the direct calculation of the final clustering outcome, improving both efficiency and effectiveness in comparison to two-step clustering methods. In this context, one-step spectral clustering techniques can be divided into two primary categories: continuous and discrete frameworks.

a) Continuous Framework: This type of one-step spectral clustering typically learns the embedding matrix so that the data points can be divided into k clusters without the need for any discretization steps. The two primary methods based on the continuous framework are as follows:

(i) **Laplacian-Based Methods:** These methods aim to learn the Laplacian matrix of a similarity graph consisting of k connected components. To achieve this, they leverage the Ky Fan's theorem alongside a fundamental principle in graph learning [19]: the multiplicity k of the eigenvalue 0 of the graph Laplacian matrix L indicates the number of connected components in the similarity graph. This relationship emphasizes the critical role of the eigenvalues of the graph Laplacian in understanding the connectivity of graph structures, facilitating more effective clustering and analysis of complex data. Inspired by this insight, a notable example of Laplacian-based methods is the CAN approach [42], whose objective function was described in Problem (8) as an adaptive neighbor method:

$$\min_{s_{ij},F} \sum_{i,j=1}^{n} (\|x_i - x_j\|_2^2 s_{ij} + \gamma s_{ij}^2) + \nu Tr(F^T L_S F) \quad (24)$$

$$\text{s.t.} \quad \sum_{j=1}^{n} s_{ij} = 1, 0 \leq s_{ij} \leq 1, F \in \mathbb{R}^{n \times k}, F^T F = I_k,$$

where ν is the regularization parameter. Here, to facilitate the learning of an optimal neighbor assignment matrix S with a distinct clustering structure, the following problem, derived from the Ky Fan's theorem for the Laplacian matrix L_S, is defined as:

$$\sum_{i=1}^{k} \lambda_i(L_s) = \min_{F \in \mathbb{R}^{n \times k}, F^T F = I_k} Tr(F^T L_S F), \quad (25)$$

where L_S is associated with $(S + S^T)/2$, and $\lambda_i(L_S)$ denotes the i-th eigenvalue of L_S, for $i = 1, \ldots, k$.

(ii) **Matrix Factorization-Based Methods:** These methods take advantage of matrix factorization techniques for their time efficiency while also utilizing spectral clustering properties to uncover non-linear structural information. A key example is the Nonnegative Embedding and Spectral Embedding (NESE) method [64], which is rooted in the relationship between symmetric nonnegative matrix factorization and spectral clustering [65]. The optimization problem for NESE is defined as follows:

$$\min_{F,H} \|D^{-\frac{1}{2}} W D^{-\frac{1}{2}} - HF^T\|_F^2 \quad (26)$$

$$\text{s.t.} \quad F, H \in \mathbb{R}^{n \times k}, F^T F = I_k, H \geq 0,$$

where F represents the spectral embedding, and H denotes the nonnegative embedding matrix that categorizes each i-th data sample into the class indicated by the position of the highest value in the i-th row of H.

b) Discrete Framework: This one-step spectral clustering method basically utilizes three techniques to learn the embedding matrix: Spectral Rotation (SR) [36], Discrete Spectral Clustering (DSC) [66], and Discrete Nonnegative Spectral Clustering (DNSC) [25].

(i) **SR-Based Methods:** SR [36] is a method designed to transform a continuous spectral embedding matrix, typically obtained from spectral clustering, into a binary matrix that more accurately represents discrete cluster indicators. Given a spectral embedding matrix F, SR seeks to determine an optimal orthogonal transformation that maps the real-valued matrix F to a binary cluster indicator matrix Y. The optimization problem for SR can be formulated as:

$$\min_{Y,R} \|F - YR^T\|_F^2 \quad (27)$$

$$\text{s.t.} \quad R \in \mathbb{R}^{k \times k}, Y \in \text{Ind}, R^T R = I_k,$$

where $Y \in \text{Ind}$ denotes the clustering outcome as a binary indicator matrix. Following this description, a basic outline for using spectral embedding and SR together is [35]:

$$\min_{F,Y,R} Tr(F^T L_{sym} F) + \|F - YR^T\|_F^2 \quad (28)$$

$$\text{s.t.} F \in \mathbb{R}^{n \times k}, R \in \mathbb{R}^{k \times k}, F^T F = I_k, Y \in \text{Ind}, R^T R = I_k.$$

(ii) **DSC-Based Methods:** These methods adopt a direct framework to compute a discrete-valued spectral embedding matrix by solving the following optimization problem:

$$\min_{F,Y} Tr(F^T L F) - \nu Tr(F^T Y) \quad (29)$$

$$\text{s.t.} \quad F \in \mathbb{R}^{n \times k}, F^T F = I_k, Y \in \text{Ind},$$

where ν denotes a regularization parameter. Notably, for any $Y \in \text{Ind}$, the optimal solution for the spectral embedding matrix is given by $F = Y(Y^T Y)^{-\frac{1}{2}}$. This approach directly derives the discrete embedding matrix F without relying on conventional relax-and-discretize methods.

(iii) **DNSC-Based Methods:** DNSC [25] is an alternative method for generating a discrete-valued spectral embedding matrix. In the absence of sign constraints on the spectral embedding matrix F, it may include both positive and negative values, deviating from the structure of a true cluster indicator matrix, where each row has exactly one positive element, with all others being zero. To resolve this, DNSC enforces both nonnegativity and orthogonality constraints on F, resulting in the following problem:

$$\min_{F} \text{Tr}(F^T L F) \quad (30)$$

$$\text{s.t.} \quad F \in \mathbb{R}^{n \times k}, F^T F = I_k, F \geq 0.$$

Essentially, by introducing a nonnegative constraint, the DNSC problem (30) becomes more consistent with the N-cut method described in (4) for producing discrete indicator vectors [67]. This improves the interpretability of the spectral embedding matrix F and eliminates the need for post-processing techniques like k-means.

IV. Category of Spectral Clustering

Spectral clustering methods are systematically categorized into single-view and multi-view approaches. Single-view spectral clustering constructs graph representations from individual data perspectives, whereas multi-view approaches integrate information from multiple data representations to achieve enhanced clustering performance. Both paradigms share three fundamental stages: graph structure learning, spectral embedding, and partition process. Multi-view spectral clustering extends this framework by incorporating two additional components: view-specific matrix construction and cross-view information fusion. This extended framework enables the discovery of intrinsic data patterns through complementary view relationships that remain undetectable in single-view analysis, leading to more comprehensive clustering solutions.

A. Single-view Spectral Clustering

Single-view spectral clustering encompasses diverse methodologies for graph-based data partitioning, distinguished by graph structure learning approaches and partitioning processes. This section presents a systematic analysis through two fundamental perspectives: graph structure paradigms and partitioning strategies. The methodology spans three principal architectures: pairwise graphs, anchor graphs, and hypergraphs. Within each, methods are categorized as fixed approaches using predetermined rules or adaptive techniques dynamically learning optimal structures. The partitioning process employs either a two-step approach with sequential spectral embedding and clustering, or a one-step approach integrating these processes into a unified optimization framework.

1) Pairwise Graph: Pairwise graphs represent relationships between data points and form the foundation of many spectral clustering algorithms. These graphs can be constructed using fixed or adaptive approaches, offering varying levels of flexibility and performance in capturing the underlying structure of the data.

Fixed Methods: Fixed methods establish graph topology through predetermined protocols, forming the foundational approach in graph construction. These methods, detailed in Table II, encompass KNN, mutual KNN, fully connected graphs, and ϵ-ball graphs. Seminal works include von Luxburg's mutual kNN formulation [19], which established theoretical foundations, and subsequent enhancements such as KNN-SC [68] for adaptive neighbor selection. Notable advancements include SR [36] for improved discretization and JSESR [35], which integrates spectral embedding and rotation in a unified framework. These approaches offer computational efficiency while maintaining consistent structural properties across diverse datasets.

Adaptive Neighbor Methods:Adaptive Neighbor Methods:

Adaptive Neighbor Methods focus on dynamically constructing graph structures directly from the relationships within the data, tailoring the clustering process to the dataset's underlying structure and enhancing the flexibility of clustering algorithms. CAN [42] pioneered this methodology by learning probabilistic neighbor connections during clustering, ensuring that the graph representation captures meaningful local relationships and global patterns in the data. PCAN [42] extended this framework to accommodate high-dimensional datasets by jointly optimizing dimensionality reduction and similarity learning, creating a unified model that improves clustering quality while reducing computational complexity. SWCAN [69] introduced adaptive feature weighting, enabling the model to prioritize features with higher relevance to clustering tasks, effectively mitigating the influence of noisy or redundant features. ERCAN [70] further refined these techniques by incorporating entropy regularization, which not only optimizes the similarity distribution but also balances the adaptiveness of neighbor assignment with robust clustering objectives, leading to improved performance on diverse datasets.

SR-OSC [71] took a step further by unifying graph construction with clustering optimization in a low-dimensional space, leveraging spectral rotation to enhance stability and accuracy. SCANDLE [72] revolutionized the field by integrating deep learning techniques, replacing traditional eigen-decomposition with neural networks, thereby improving scalability and effectiveness for large and complex datasets. PTAG [73] introduced a groundbreaking typicality-based neighbor assignment framework, which eliminates the conventional sum-to-one constraints in similarity assignment and incorporates block-diagonal regularization. This innovation strengthens cluster separability and provides increased robustness against noise and outliers. These advancements, summarized in Table II, underscore the evolution of spectral clustering toward adaptive and efficient single-step frameworks capable of handling large- and high-dimensional data while maintaining precision and robustness.

Initial Similarity Matrix Methods: Initial similarity matrix methods refine pre-constructed graphs to improve clustering performance. These approaches have gained prominence for their ability to directly optimize graph structures for clustering tasks. The development of these methods can be traced through several key algorithms. The Constrained Laplacian Rank (CLR) algorithm [44] learns a new similarity matrix with exactly k connected components from an initial similarity matrix, employing a one-step continuous approach through Ky Fan theory to reveal cluster structure without post-processing. Building upon CLR, RCFE [74] enhances spectral clustering by simultaneously learning adaptive graph structure and low-dimensional embedding, using Laplacian rank constraints to recover block-diagonal similarity matrices. For high-dimensional data, RCFE implements a flexible embedding scheme allowing mismatch between projected and embedded features. SClump [75] addresses heterogeneous information networks by integrating multiple meta-path similarities, optimizing similarity matrices and meta-path weights simultaneously. The method employs Laplacian rank constraints with iterative refinement. More recently, FCLR [76] accelerated CLR using anchors and bipartite graph learning strategies, reducing computational complexity from quadratic to linear with respect to data size.

Self-expressive Methods: Self-expressive methods construct similarity matrices by representing each data point as

TABLE II: Comparison of Fixed, Adaptive Neighbor, and Initial Similarity Matrix Methods for Spectral Clustering

#	Method	Graph Structure Learning	Two-Step	One-Step (Continuous)	One-Step (Discretized)
1	MKNNG (2007) [19]	Mutual-KNN	✓		
2	SRC (2013) [36]	KNN			✓
3	JSESR (2020) [35]	KNN			✓
4	KNN-SC (2021) [68]	KNN	✓		
5	CAN (2014) [42]	$\min_{s_{ij}} \sum_{i,j=1}^{n} (\|x_i - x_j\|_2^2 s_{ij} + \gamma s_{ij}^2)$		✓	
6	PCAN (2014) [42]	$\min_{s_{ij},P} \sum_{i,j=1}^{n} (\|P^T x_i - P^T x_j\|_2^2 s_{ij} + \gamma s_{ij}^2)$		✓	
7	SEANC (2019) [77]	$\min_{s_{ij}} \sum_{i,j=1}^{n} (\|f_i - f_j\|_2^2 s_{ij} + \gamma s_{ij}^2 + \beta \|g_i - g_j\|_2^2 s_{ij})$		✓	
8	SR-OSC (2020) [71]	$\min_{s_{ij},P} \sum_{i,j=1}^{n} (\|P^T x_i - P^T x_j\|_2^2 s_{ij} + \gamma s_{ij}^2)$			✓
9	SWCAN (2020) [69]	$\min_{s_{ij},\Theta} \sum_{i,j=1}^{n} (\|\Theta x_i - \Theta x_j\|_2^2 s_{ij} + \gamma s_{ij}^2)$		✓	
10	ERCAN (2022) [70]	$\min_{s_{ij}} \sum_{i,j=1}^{n} (\|x_i - x_j\|_2^2 s_{ij} + \gamma s_{ij} \log s_{ij})$		✓	
11	SCANDLE (2023) [72]	$\min_{s_{ij}} \sum_{i,j=1}^{n} (\|x_i - x_j\|_2^2 s_{ij} + \gamma s_{ij}^2)$	✓		
12	CTAG (2023) [73]	$\min_{s_{ij}} \sum_{i,j=1}^{n} \|x_i - x_j\|_2^2 s_{ij} + \sum_{i=1}^{n} \gamma_i \sum_{j=1}^{n} (s_{ij} \log s_{ij} - s_{ij})$		✓	
13	CLR (2016) [44]	$\min_{S} \|S - A\|_F^2$		✓	
14	RCFE (2018) [74]	$\min_{S} \|S - A\|_F^2 + \alpha \|X^T P - F\|_F^2$		✓	
15	SClump (2019) [75]	$\min_{S} \|S - A\|_F^2 + \alpha \|S\|_F^2$		✓	
16	FCLR (2024) [76]	$\min_{S} \|S - A\|_F^2$		✓	

a linear combination of others in the dataset. This results in a coefficient matrix that encodes intrinsic relationships and subspace structures, effectively capturing the underlying data distribution.

Sparse Subspace Clustering (SSC) [45] pioneered the use of ℓ_1-norm minimization to learn a sparse coefficient matrix. This method is particularly effective in dealing with data derived from multiple linear subspaces by ensuring that each data point is only connected to points within its own subspace, achieving high precision in clustering. Low-Rank Representation (LRR) [78] extended this framework by employing nuclear norm minimization to obtain a low-rank representation. This approach excels at capturing global structural information while demonstrating robustness to noise and outliers, making it suitable for diverse real-world applications. Least Squares Regression (LSR) [79] introduced Frobenius norm minimization, which simplified the optimization process through a closed-form solution. This not only reduced computational complexity but also maintained high efficiency, making it a practical choice for large-scale datasets.

Smooth Representation (SMR) [80] enhanced clustering performance by incorporating a smoothness term derived from a k-nearest neighbor graph. This innovation preserved local structural information while enforcing a grouping effect among similar data points, ensuring better consistency in the generated clusters. Laplacian Regularized LRR (LapLRR) [81] integrated manifold regularization into the low-rank representation framework. By simultaneously preserving global Euclidean structure and local manifold properties, this method effectively balanced global and local data characteristics for improved clustering accuracy. Dual Graph Regularized LRR (DGLRR) [82] further advanced this idea by introducing dual graph regularization to model both data and feature manifolds. This dual perspective captured a more comprehensive view of the relationships within the data, improving clustering performance on complex datasets.

Recent advancements have focused on leveraging deep learning and adaptive techniques to enhance clustering. Graph Compact LRR (GCLRR) [83] combined dictionary learning with clustering, modeling the dictionary as a linear combination of data points to achieve compact and efficient representations. Deep Subspace Clustering Networks (DSC-Nets) [84] utilized self-expressive layers within autoencoder architectures, enabling the learning of nonlinear data mappings and achieving significant improvements in clustering for nonlinear subspaces. Similarly, Structured AutoEncoders (StructAE) [85] incorporated self-expressive layers while adding structural constraints, further boosting clustering accuracy for complex datasets. LRR with Adaptive Graph Regularization (LRR-AGR) [86] improved graph construction by incorporating distance regularization and enforcing non-negative constraints, adapting dynamically to the data for better clustering results.

Low-Rank Sparse Subspace (LSS) [87] proposed a dynamic approach for learning similarity matrices directly from low-dimensional spaces, enhancing its ability to identify intrinsic structures within the data. Subspace Clustering via Iterative Discrete Optimization (SSC-ID) [88] introduced a robust one-step clustering method that employed spectral rotation and self-paced learning, ensuring resilience against noisy data and enhancing clustering quality. The latest development, Subspace Clustering via Joint $\ell_{1,2}$ and $\ell_{2,1}$ norms (SCJL12-21) [89], tackled the challenge of heavily corrupted data by applying joint norm minimization, effectively excluding corrupted data points while maintaining robustness to outliers.

Table III provides a detailed comparison of these methods, outlining their graph construction strategies and clustering steps.

2) Anchor Graph: Anchor Graph methods have significantly enhanced spectral clustering for large-scale data by replacing the full similarity matrix with a more efficient representation based on anchor points. This innovation reduces

TABLE III: Comparison of Self-Expressive Methods Spectral Clustering Methods: Graph Construction and Clustering Steps

#	Method	Graph Structure Learning		Partitioning Process		
				Two-Step	One-Step	
		Loss(X, Z)	Regularization		Continuous	Discretized
1	SSC (2012) [45]	$\|X - XZ\|_1$	$\|Z\|_1$	✓		
2	LRR (2013) [78]	$\|X - XZ\|_{21}$	$\|Z\|_*$	✓		
3	LSR (2013) [79]	$\|X - XZ\|_F$	$\|Z\|_F^2$	✓		
4	SMR (2014) [80]	$\|X - XZ\|_{21}$	$\text{tr}(ZLZ^T)$	✓		
5	LapLRR (2014) [81]	$\|X - XZ\|_F$	$\|Z\|_* + \text{tr}(ZLZ^T)$	✓		
6	DGLRR (2015) [82]	$\|X - XZ - GX\|_1$	$\|Z\|_* + \|G\|_* + \text{tr}(ZL_ZZ^T) + \text{tr}(GL_GG^T)$	✓		
7	GCLRR (2017) [83]	$\|X - XZ\|_{21}$	$\|Z\|_* + \text{tr}(ZLZ^T)$	✓		
8	DSC-Nets (2017) [84]	$\|X - \hat{X}\|_F^2 + \|H - HZ\|_F^2$	$\|Z\|_p$	✓		
9	StructAE (2018) [85]	$\|X - H^{(M)}\|_F^2 + \|H^{(M/2)} - H^{(M/2)}Z\|_F^2$	$\sum_{m=1}^M(\|W^{(m)}\|_F^2 + \|b^{(m)}\|_2^2)$	✓		
10	LRR-AGR (2018) [86]	$\|X - XZ\|_1$	$\|Z\|_*$		✓	
11	LSS (2019) [87]	$\|P^TX - P^TXZ^{(1)}\|_F^2 + \|X - XZ^{(2)}\|_F^2$	$\sum_{i=1}^n(\|Z_i^{(1)} - Z_i^{(2)}\|_2^2 + \|P\|_{21})$		✓	
12	SSC-ID (2020) [88]	$\|X - XZ\|_F^2$	$\|Z\|_1$			✓
13	SCJL12-21 (2024) [89]	$\|X - XZ\|_{21}$	$\|Z\|_{12}$	✓		

computational complexity from $O(n^2)$ or $O(n^3)$ to $O(nm)$, where n represents the number of data points and m is the number of anchors ($m \ll n$).

LSC [49] introduced k-means clustering for anchor selection and employed a fixed Gaussian kernel for graph construction, following a two-step clustering process. FSC [90] advanced the field by adopting BKHK for anchor selection and introducing an adaptive, parameter-free approach for graph construction, while retaining the two-step process. FSCAG [91] utilized random anchor selection combined with adaptive graph construction and transitioned to a continuous one-step clustering framework. SGCNR [92] applied k-means for anchor selection alongside adaptive graph construction and incorporated nonnegative relaxation within a continuous one-step clustering model. RLGCH [93] implemented k-means++ for anchor selection and a fixed Gaussian kernel, employing a discretized one-step clustering method enhanced by the $\ell_{2,1}$-norm to improve robustness.

FGC-SS [94] utilized BKHK for anchor selection with adaptive graph construction, unifying spectral embedding and rotation within a discretized one-step framework. EDCAG [48] introduced BKHK for anchor selection and adaptive graph construction while leveraging a bipartite similarity matrix in a discretized one-step clustering process. RESKM [95] reverted to k-means for anchor selection and employed fixed graph construction, maintaining the traditional two-step clustering process. FSBGL [51] employed k-means clustering for anchor selection, combined with adaptive graph construction, and utilized a bipartite similarity matrix within a continuous one-step framework. SE-ISR [96] also adopted k-means for anchor selection, integrating adaptive graph construction and unifying spectral embedding with rotation in a discretized one-step process. SCLE [97] applied BKHK for anchor selection and adaptive graph construction, introducing linear embedding into a discretized one-step clustering framework.

3) Hypergraph Methods: Hypergraph spectral clustering methods encompass three main approaches: hypergraph Laplacians [53], hypergraph reduction [98], and tensor modeling [99]. Hypergraph Laplacian methods generalize graph Laplacians to higher-order data structures, preserving the original hypergraph topology. Hypergraph reduction transforms hypergraphs into standard graphs through methods such as Star-Expansion [100], Clique-Expansion [101], and Line-Expansion [102]. Tensor modeling represents uniform hypergraphs as higher-dimensional tensors [99]. This survey primarily focuses on hypergraph Laplacian methods, which yield accurate results for complex, high-order relationships while avoiding information loss associated with reduction techniques. These methods effectively capture intrinsic multi-way relationships in real-world datasets by operating directly on the hypergraph structure, although they come with higher computational costs.

Hypergraph methods can be broadly categorized into fixed and adaptive construction approaches. Among fixed construction methods, Normalized Hypergraph Spectral Clustering (NHSC) [52] extends spectral clustering to hypergraphs using k-nearest neighbor (KNN) hypergraph construction. Unsupervised Image Categorization (UIC) [103] also employs KNN-based hypergraph construction for image analysis. Context-Aware Hypergraph Spectral Clustering (CAHSM) [104] combines multiple similarity measures within a fixed hypergraph structure to enhance clustering results. Hypergraph Spectral Clustering with Local Refinement (HSCLR) [30] uses a k-means-based hypergraph construction for improved accuracy. More recent fixed approaches include Hypergraph Spectral Clustering (HSC) for point cloud segmentation [29] and GraphLSHC [105], both of which utilize KNN-based hypergraph construction for efficient processing.

Adaptive construction methods provide more flexibility in capturing data relationships, allowing them to better handle complex patterns. The Regression-based Hypergraph (RH) [106] method employs sparse representation to construct hypergraphs, facilitating more precise clustering. Elastic-Net Hypergraph (ENHG) [60] uses elastic net regularization to improve the stability and accuracy of the constructed hypergraph. Correntropy-Induced Robust Low-Rank Hypergraph (CIRLH) [107] combines low-rank representation with a correntropy-induced metric, providing robustness to noise. The Robust ℓ_2-Hypergraph (RHℓ_2) [59] employs affine subspace ridge regression for better clustering performance under noisy conditions. More recent adaptive methods, such as Elastic-Net constrained Hypergraph (EN-cHG) [108] and Hessian Non-negative Hypergraph (HNH) [109], integrate various regularization techniques to enhance clustering performance and improve robustness against high-dimensional data and noise.

Table V provides a comprehensive comparison of these

TABLE IV: Comparison of Anchor Graph Methods Spectral Clustering Methods: Graph Construction and Clustering Steps

#	Method	Graph Structure Learning			Partitioning Process		
					Two-Step	One-Step	
		Select Anchor	Anchor Graph	Similarity Matrix		Continuous	Discretized
1	LSC (2015) [49]	K-means	Fixed	Gram	✓		
2	FSC (2017) [90]	BKHK	Adaptive	Gram	✓		
3	FSCAG (2017) [91]	Random	Adaptive	Gram		✓	
4	SGCNR (2019) [92]	K-means	Adaptive	Gram		✓	
5	RLGCH (2022) [93]	K-means++	Fixed	Gram			✓
6	FGC-SS (2022) [94]	BKHK	Adaptive	Gram			✓
7	EDCAG (2023) [48]	BKHK	Adaptive	Bipartite			✓
8	RESKM (2023) [95]	K-means	Fixed	Gram	✓		
9	FSBGL (2023) [51]	K-means	Adaptive	Bipartite		✓	
10	SE-ISR (2023) [96]	K-means	Adaptive	Gram			✓
11	SCLE (2024) [97]	BKHK	Adaptive	Gram			✓

hypergraph-based spectral clustering methods, highlighting their graph construction approaches and clustering steps. As shown in the table, all methods employ a two-step clustering approach, regardless of the construction method used. The transition from fixed to adaptive construction techniques reflects the ongoing effort to enhance hypergraph learning's ability to capture complex data relationships, address challenges like noise, and handle high-dimensional data, thereby improving clustering performance in real-world applications.

TABLE V: Comparison of Hypergraph-Based Spectral Clustering Methods: Graph Construction and Clustering Steps

#	Method	Graph Structure Learning	Partitioning Process		
			Two-Step	One-Step	
				Continuous	Discretized
1	NHSC (2006) [52]	KNN-HG	✓		
2	UIC (2011) [103]	KNN-HG	✓		
3	CAHSM (2014) [104]	KNN-HG	✓		
4	HSCLR (2018) [30]	K-means-HG	✓		
5	HSC (2020) [29]	KNN-HG	✓		
6	GLSHC (2021) [105]	KNN-HG	✓		
7	RH (2016) [106]	Sparse Representation	✓		
8	ENHG (2017) [60]	EN-HG	✓		
9	CIRLH (2019) [107]	Correntropy-Induced	✓		
10	RHL_2 (2019) [59]	L2-HG	✓		
11	EN-cHG (2020) [108]	EN-HG	✓		
12	HNH (2023) [109]	HN-HG	✓		

B. Multi-view Spectral Clustering

Multi-view spectral clustering represents a significant advancement over traditional spectral clustering by incorporating multiple complementary representations of data [110]. This methodology addresses scenarios where data exhibits heterogeneous characteristics captured through diverse feature sets, measurement modalities, or representation schemes [111]. For example, multimedia content analysis often generates multiple natural views, such as visual, textural, and semantic features. By simultaneously leveraging these multiple perspectives, multi-view clustering demonstrates superior robustness and accuracy compared to single-view approaches [112].

The multi-view framework extends the conventional spectral clustering paradigm by introducing sophisticated matrix construction and graph fusion mechanisms while preserving the fundamental stages of graph learning, spectral embedding, and partition optimization. Information fusion in multi-view spectral clustering encompasses four principal strategies: (1) early fusion, which integrates view information prior to clustering [113]–[116], (2) late fusion, which combines individual view clustering outcomes [117], (3) hybrid fusion, which integrates information at multiple stages [118], and (4) score fusion, which aggregates view-specific clustering probabilities [119].

The choice of fusion method is critically dependent on the specific requirements of the clustering task, the nature of the multi-view data, and the desired balance between computational efficiency and clustering accuracy. Early fusion approaches, which create a cohesive representation of the data before applying clustering algorithms, often capture more intricate relationships between views and are the primary focus of many recent studies. With the fusion strategy determined, the next crucial step in multi-view spectral clustering is the construction of appropriate matrices to represent the fused data.

a) Matrix Construction: In multi-view spectral clustering, matrix construction offers two primary options - similarity matrix, directly constructed from the graph to represent pairwise similarities between data points across views, and Laplacian matrix, which encodes both similarity and degree information.

b) Graph Fusion: Graph fusion, a critical step in multi-view spectral clustering, integrates information from multiple views into a unified representation. It comprises three main components:

- Consensus Information [120]: Identifies and emphasizes common patterns across different views, enhancing model robustness and generalization capability.
- Complementary Information [121]: Leverages unique insights provided by each view, enriching the overall representation with diverse perspectives.
- View Weighting [122]: Assigns appropriate weights to each view, crucial for effective fusion. Strategies include equal-weighted, auto-weighted (parameter-weighted or parameter-free), and advanced approaches considering feature and sample weights, confidence levels, and cluster-wise weighting.
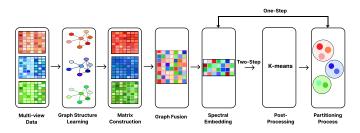

Fig. 4: Multi-view Spectral Clustering Framework.

Figure 4 presents a systematic framework for multi-view spectral clustering, illustrating the sequential processes of graph construction, matrix formulation, information fusion, spectral embedding, and partition optimization. Building upon these foundational elements of multi-view spectral clustering, particularly the fusion mechanisms and matrix construction strategies, subsequent sections examine three principal graph learning paradigms: pairwise graphs, anchor-based representations, and hypergraph structures.

1) Pairwise Graph: Pairwise graph methods in multi-view spectral clustering focus on learning similarity graphs that capture cross-view relationships. Following the principles established in single-view clustering, these methods are categorized into fixed and adaptive approaches. While fixed methods employ predetermined rules, adaptive methods encompass three principal strategies: adaptive neighbor , initial similairy graph , and self-expressive learning. The subsequent sections analyze representative algorithms from each category, with self-expressive models examined separately due to their distinct theoretical foundations. Table VI presents a systematic comparison of these methods, highlighting their graph learning mechanisms and fusion strategies.

Fixed Methods: Fixed methods use a predetermined number of neighbors for each data point. This category includes several representative methods. S-MVSC [113] uses KNN graphs and learns a sparse consensus similarity matrix directly without iterations. It doesn't model complementary information or view weights. The approach is two-step: first fusing graphs, then clustering with k-means. Its main advantage is speed and simplicity due to the closed-form solution. AMGL [114] fuses Laplacian matrices from multiple views using automatically learned weights. It alternates between updating the fused Laplacian and the cluster indicator matrix, then applies k-means for final clustering. This parameter-free approach offers a simple yet effective solution for multi-view spectral clustering. SwMC [115] learns a consensus graph from multiple views using Constrained Laplacian Rank. It automatically determines view weights without additional parameters. The method directly obtains clusters from the learned graph using Ky Fan theory, eliminating the need for post-processing like k-means. AMCSE [123] integrates multi-view graph learning, spectral embedding, and clustering into a unified framework. It adaptively weights view-specific Laplacian matrices, fuses them, and uses spectral rotation for direct discretization. This one-step approach jointly optimizes graph construction, view importance, and clustering, iteratively refining spectral embeddings and cluster assignments to improve multi-view clustering performance.

Adaptive Neighbor Methods: Adaptive neighbor methods dynamically adjust the neighborhood size based on data characteristics. MLAN [116] is a graph-based multi-view learning method that adaptively constructs a consensus similarity matrix across views. It automatically learns view weights without parameters, alternately optimizing the graph and weights. The final graph directly yields clusters via connected components, avoiding k-means and offering a parameter-free approach for clustering. GBS [124] fuses multi-view similarity graphs into a consensus graph using auto-weighted fusion. It applies Ky

Fan's theorem to constrain the Laplacian matrix, enabling direct one-step clustering without post-processing. This approach automatically weights views and produces clusters from the consensus graph.

GMC [125] fuses multi-view similarity graphs into a unified graph, jointly optimizing view-specific and consensus graphs. It uses auto-weighted fusion and applies Ky Fan's theorem to constrain the Laplacian matrix, enabling direct one-step clustering. This approach mutually reinforces individual view graphs and the consensus graph, automatically weighting views without requiring post-processing. MVCGE [126] integrates multi-view consensus graph learning, spectral embedding, and nonnegative embedding (NESE) into a unified framework. It jointly optimizes these components using kernel matrices from multiple views, employing spectral convolution and automatic view weighting. This one-step approach aims to improve multi-view clustering by simultaneously refining graph structure, spectral representation, and cluster assignment without post-processing.

SCGLD [127] integrates consensus graph learning and discrete clustering for multi-view data in a unified framework. It adaptively learns a shared graph across views while directly obtaining discrete cluster labels, avoiding information loss from separate steps. This one-step approach iteratively refines the consensus graph and cluster indicators, allowing them to mutually enhance each other, aiming to improve multi-view clustering performance. CSRF [128] method fuses multi-view information at the spectral embedding level. It learns a consensus low-dimensional embedding by exploiting spectral rotation to combine individual view embeddings, reducing redundancy and capturing complementary information. CSRF uses an $\mathcal{O}(n^2)$ alternating optimization algorithm, incorporates flexible graph construction schemes, and constructs a fused similarity matrix from the consensus embedding for final clustering. This approach aims to improve multi-view clustering efficiency and effectiveness compared to methods operating on raw data or fixed graphs.

AGLSR [129] is a one-step approach for multi-view clustering. It adaptively learns similarity graphs for each view, designs a shared spectral embedding across views, and uses spectral rotation to obtain clustering labels directly. AGLSR integrates graph learning, fusion, and partitioning into a unified framework, avoiding the need for separate k-means clustering on a consensus graph. This approach aims to improve clustering performance by closely unifying graph learning with partition generation.

Self-expressive Methods: Early developments in self-expressive multi-view spectral clustering focused on foundational representation learning and noise handling. As shown in Table VII, these methods can be analyzed through their graph structure learning, fusion strategies, and partitioning processes. DiMSC [130] establishes a comprehensive framework integrating graph construction and fusion. It employs smooth representation with Frobenius norm $|X^v - X^v Z^v|_F^2$ and utilizes graph Laplacian regularization $\mathrm{Tr}(Z^v L^v Z^{vT})$. Through the Hilbert-Schmidt Independence Criterion, it enhances view diversity and complementarity, providing efficient optimization with theoretical convergence guarantees via Sylvester equations.

TABLE VI: Comparison of Fixed, Adaptive Neighbor, and Initial Similarity Matrix Multi-view Spectral Clustering Methods: Graph Construction and Clustering Steps

#	Method	Graph Structure Learning	Graph Fusion			Partitioning Process		
			Consensus	Complementary	Weight	Two-Step	One-Step Continuous	Discretized
1	AMGL (2016) [114]	KNN			✓	✓		
2	S-MVSC (2018) [113]	KNN	✓			✓		
3	SwMC (2017) [115]	KNN	✓				✓	
4	AMCSE (2020) [123]	KNN			✓			✓
5	MLAN (2018) [116]	$\min_{s_{ij}} \sum_{v=1}^{V} \alpha_v \sum_{i,j=1}^{n}(\|x_i^v - x_j^v\|_2^2 s_{ij} + \gamma s_{ij}^2)$			✓		✓	
6	MVGL (2018)	$\min_{s_{ij}} \sum_{v=1}^{V} \alpha_v \sum_{i,j=1}^{n}(\|x_i - x_j\|_2^2 s_{ij} + \gamma s_{ij}^2)$	✓		✓		✓	
7	OMSC (2019)	$\min_{s_{ij},t_{ij},P,f} \sum_{i,j=1}^{n}\left(\|P^T x_i - P^T x_j^v\|_2^2 s_{ij} + \|f_i - f_j\|^2 t_{i,j} + \|s_i^{(v)} - t_i\|^2\right)$			✓		✓	
8	GBS (2019) [124]	$\min_{s_{ij}} \sum_{v=1}^{V} \alpha_v \sum_{i,j=1}^{n}(\|x_i^v - x_j^v\|_2^2 s_{ij} + \gamma s_{ij}^2)$	✓		✓		✓	
9	GMC (2020) [125]	$\min_{s_{ij},P} \sum_{v=1}^{V} \alpha_v \sum_{i,j=1}^{n}(\|P^T x_i^v - P^T x_j^v\|_2^2 s_{ij} + \gamma s_{ij}^2)$	✓		✓		✓	
10	MVCGE (2022) [126]	$\min_{s_{ij},P} \sum_{v=1}^{V} \alpha_v \sum_{i,j=1}^{n}(\|P^T x_i^v - P^T x_j^v\|_2^2 s_{ij} + \gamma s_{ij}^2)$	✓		✓		✓	
11	SCGLD (2022) [127]	$\min_{s_{ij}} \sum_{v=1}^{V} \alpha_v \sum_{i,j=1}^{n}(\|x_i^v - x_j^v\|_2^2 s_{ij} + \gamma s_{ij}^2)$	✓		✓			✓
12	CSRF (2023) [128]	$\min_{s_{ij}} \sum_{v=1}^{V} \alpha_v \sum_{i,j=1}^{n}(\|x_i^v - x_j^v\|_2^2 s_{ij} + \gamma s_{ij}^2)$	✓			✓		✓
13	AGLSR (2023) [129]	$\min_{s_{ij}} \sum_{v=1}^{V} \alpha_v \sum_{i,j=1}^{n}(\|x_i^v - x_j^v\|_2^2 s_{ij} + \gamma s_{ij}^2)$			✓			✓

IMVSC [131] advances this through a novel iterative framework combining multi-graph Laplacian regularization with low-rank and sparsity constraints. The method employs composite regularization terms $\|Z^v\|_* + \|Z^v\|_1 + \mathrm{Tr}(Z^v L^v Z^{vT})$ with a robust ℓ_1-norm reconstruction loss $|X^v - X^v Z^v|_1$. This approach effectively preserves view-specific structures while promoting cross-view consensus through alternating optimizationoptimization.

ECMSC [132] introduces position-aware exclusivity terms with ℓ_1-norm graph construction $\|Z^v\|_1$, enhancing complementary information while maintaining common structures. MLRSSC [133] further advances subspace learning by simultaneously imposing nuclear and ℓ_1-norm regularization $|Z^v| + \|Z^v\|_1$ with consensus constraints to learn a shared affinity matrix. CSMSC [134] proposes a novel unified model exploiting both consistency and specificity through shared and view-specific representations $(C + D^v)$. Using $\ell_{2,1}$-norm reconstruction error $|X^v - X^v(C + D^v)|_{21}$ and composite regularization $|C|_* + |D^v|_F^2$, it effectively captures common properties and inherent differences across views. FMR [135] advances flexible representation learning using nuclear-norm regularization $|Z^v|$ with weighted HSIC measures, capturing nonlinear relationships while avoiding single-view limitations. GFSC [136] introduces adaptive graph fusion with Frobenius norm regularization $|X^v - X^v Z^v|_F^2 + \alpha|Z^v|_F^2$, maintaining optimal clustering properties through explicit structure constraints.

Recent developments have focused on sophisticated integration of local-global structures and advanced fusion strategies. MSCCSM [137] maximizes the common structure between local and global representations through a dynamic learning framework. The method employs ℓ_2-norm regularization $|W|_F^2$ combined with global reconstruction loss $|X^v - X^v Z|_F^2$, while utilizing a structure fusion term $|Z \odot W|_1$ to enhance view complementarity. This unified approach enables better preservation of both local manifold structure and global clustering information. DMSC-UDL [138] represents a significant advancement by leveraging deep neural networks for graph construction. The method uniquely combines manifold structure preservation through $\mathrm{Tr}(H^{vT} L H^v)$ with discriminative cross-view constraints $|S_i \otimes S_j|_1$. This deep learning framework effectively captures nonlinear relationships while maintaining complementary information across views, leading to more discriminative clustering results.

COMVSC [139] introduces a novel one-step framework that unifies affinity learning, partition fusion, and clustering. Through subspace representation construction $|X^v - X^v Z^v|_F^2 + \alpha|Z^v|_F^2$ and partition-level fusion, it avoids the limitations of noisy affinity matrices. Notably, it employs spectral rotation for direct cluster assignment, eliminating the need for additional post-processing steps and ensuring more coherent clustering results. MVSCGE [140] advances locality preservation in learned subspaces through a comprehensive approach combining grouping effects with cross-view consistency. The method's smooth regularization $\lambda \mathrm{tr}(Z^{(v)} L^{(v)} Z^{(v)^T})$ ensures robust local structure preservation, while its unified optimization framework jointly learns subspace representations and consistent cluster indicators across views, effectively balancing local and global information. MAGC [141] presents an innovative framework integrating multiple graph structures and node attributes through sophisticated graph filtering. Unlike deep learning approaches, it employs efficient graph filtering $|X^v - X^v S|_F^2$ combined with weighted high-order random walk proximity $|Z - f(A^v)|_F^2$. This design enables smooth node representations while adaptively learning consensus relationships across heterogeneous views. The method's flexible regularizer captures complex relationships across views, making it particularly effective for handling multi-view attributed graphs.

2) Anchor Graph: Anchor graph-based methods in multi-view spectral clustering offer a significant advancement in managing scalability challenges. By using a small set of representative points (anchors), these methods approximate the full similarity structure, reducing computational complexity while preserving critical data relationships. MVSC [142] selects anchors through k-means on concatenated features. It employs a fixed method to generate anchor graphs and constructs bipartite graphs for similarity matrices. The method uses a weighted combination for view fusion and performs spectral embedding on the fused graph to obtain low-dimensional

TABLE VII: Comparison of Self-expressive Multi-view Spectral Clustering Methods: Graph Construction and Clustering Steps

#	Method	Graph Structure Learning		Graph Fusion			Partitioning Process		
		Loss(X, Z)	Regularization	Consensus	Complementary	Weight	Two-Step	One-Step	
								Continuous	Discretized
1	DiMSC (2015) [130]	$\|X^v - X^v Z^v\|_F^2$	$Tr(Z^v L^v Z^{vT})$		✓	✓	✓		
2	IMVSC (2016) [131]	$\|X^v - X^v Z^v\|_1$	$\|Z^v\|_* + \|Z^v\|_1 + Tr(Z^v L^v Z^{vT})$	✓			✓		
3	ECMSC (2017) [132]	$\|X^v - X^v Z^v\|_1$	$\|Z^v\|_1$	✓	✓		✓		
4	MLRSSC (2018) [133]	$\|X^v - X^v Z^v\|_F^2$	$\|Z^v\|_* + \|Z^v\|_1$	✓			✓		
5	CSMSC (2018) [134]	$\|X^v - X^v(C + D^v)\|_{21}$	$\|C\|_* + \|D^v\|_F^2$	✓	✓		✓		
6	FMR (2019) [135]	$\|X^v - X^v Z^v\|_{21}$	$\|Z^v\|_*$		✓		✓		
7	GFSC (2020) [136]	$\|X^v - X^v Z^v\|_F^2$	$\|Z^v\|_F^2$	✓			✓		
8	MSCCSM (2021) [137]	$\|X^v - X^v Z^v\|_F^2$	$\|Z^v\|_F^2$	✓			✓		
9	DMSC-UDL (2021) [138]	$\|X^v - \tilde{X}^v\|_F^2 + \|H^v - H^v Z\|_F^2$	$Tr(H^{vT} L H^v) + \|Z^v\|_F^2$		✓			✓	
10	COMVSC (2022) [139]	$\|X^v - X^v Z^v\|_F^2$	$\|Z^v\|_F^2$	✓		✓			✓
11	MvSCGE (2022) [140]	$\|X^v - X^v Z^v\|_F^2$	$Tr(Z^v L^v Z^{vT})$	✓			✓		
12	MAGC (2023) [141]	$\|X^v - X^v Z\|_F^2$	$\|Z - f(A^v)\|_F^2$			✓	✓		

representations, followed by k-means clustering to derive the final data partitions. DSC [66] adopts k-means for anchor selection and constructs adaptive anchor graphs. It generates similarity matrices via bipartite graphs, incorporates view-specific weights automatically, and achieves clustering through a single-step, continuous optimization process that directly learns a consistent partition across views.

FMDC [27] utilizes k-means++ for anchor point selection and adopts an adaptive method for anchor graph generation. The similarity matrices are computed using the Gram matrix approach. Views are combined using learned weights, and clustering is achieved through a single-step, discretized optimization process that directly learns the cluster indicator matrix across views. SFMC [143] selects anchor points using k-means++ and employs an adaptive method for anchor graph construction. It generates similarity matrices through bipartite graphs and achieves clustering in a single-step, continuous optimization process, directly learning a consistent partition across views using Ky Fan's theorem. Similarly, BIGMC [144] selects anchors via k-means and constructs adaptive anchor graphs using bipartite graphs between data points and anchors. It incorporates consensus information, learns view-specific weights automatically, and performs clustering through a single-step, continuous optimization process, directly learning a joint graph compatible across views.

ERMC-AGR [28] and ECMC [28] both use k-means for anchor selection and fixed methods for anchor graph construction. ERMC-AGR computes similarity matrices using the Gram matrix approach, incorporates consensus information across views, and merges views with learned weights. The clustering process involves two stages: spectral embedding on the fused graph followed by k-means clustering for final partitioning. On the other hand, ECMC generates similarity matrices using bipartite graphs and automatically assigns weights during view fusion. It achieves clustering through a single-step, discretized optimization process that directly learns the cluster indicator matrix. FMCSE [145] also utilizes k-means for anchor point selection but adopts an adaptive method for anchor graph generation. Similarity matrices are computed using the Gram matrix approach, and views are combined with learned weights. The method performs clustering through a single-step, discretized optimization process, directly learning the cluster indicator matrix.

E^2OMVC [48] and FMAGC [146] follow similar methodologies with slight variations. E^2OMVC employs k-means for anchor selection, uses a fixed method for anchor graph construction, and generates similarity matrices using the Gram matrix approach. It incorporates consensus information across views, merges views using learned weights, and performs clustering through a single-step, discretized optimization process that directly learns the cluster indicator matrix. FMAGC uses k-means++ for anchor selection, employs an adaptive method for anchor graph construction, and computes similarity matrices using the Gram matrix approach. It incorporates consensus information across views and combines them through a weighted approach. The clustering process involves a single-step, discretized optimization that directly learns the cluster indicator matrix. Table VIII provides a comparative overview of these methods, summarizing their approaches to graph construction and clustering steps.

3) Hypergraph: Hypergraph-based methods for multi-view spectral clustering remain underexplored, with limited research addressing this promising direction. Hao et al. [147] introduced a hypergraph-based method that adaptively weighs views while capturing higher-order relationships. The approach constructs hypergraph Laplacians to encode high-order relations and employs a self-weighting strategy to determine view importance. By integrating hypergraph regularization with self-weighting in a self-representation framework, it learns an optimal consensus representation, preserving both view-specific and high-order information, followed by spectral clustering for partitioning. Yang et al. [148] proposed an adaptive hypergraph learning method to model higher-order relationships in multi-view data. This method learns view-specific similarity matrices and balances shared and view-specific information using a weighted fusion strategy, leading to improved clustering via a two-step process. Chen et al. [149] developed IMVC-HG for incomplete multi-view clustering. The method generates view-specific affinity matrices through self-representation, constructs a consistent hypergraph, and integrates representation learning and clustering using non-negative matrix factorization with orthogonal constraints in a unified framework, capturing both consensus and view-specific information.

TABLE VIII: Comparison of Anchor Graph Multi-view Spectral Clustering Methods: Graph Construction and Clustering Steps

#	Method	Graph Structure Learning			Graph Fusion			Partitioning Process		
								Two-Step	One-Step	
		Select Anchor	Anchor Graph	Similarity Matrix	Consensus	Complementary	Weight		Continuous	Discretized
1	MVSC (2015) [142]	K-means	Fixed	Bipartite			✓			
2	DSC (2019) [66]	K-means	Fixed	Gram			✓			✓
3	FMDC (2021) [27]	K-means++	Adaptive	Gram			✓			✓
4	SFMC (2022) [143]	K-means	Fixed	Bipartite			✓		✓	
5	BIGMC (2022) [144]	K-means	Adaptive	Bipartite	✓		✓		✓	
6	ERMC-AGR (2022) [28]	K-means	Fixed	Gram	✓		✓	✓		✓
7	ECMC (2022) [28]	K-means	Fixed	Bipartite	✓		✓			✓
8	FMCSE (2022) [145]	K-means	Adaptive	Gram			✓			✓
9	E^{2}OMVC (2023) [48]	K-means	Fixed	Gram	✓		✓			✓
10	FMAGC (2024) [146]	K-means++	Adaptive	Gram			✓			✓

V. FUTURE RESEARCH DIRECTION

The field of single-view and multi-view spectral clustering has seen remarkable advancements in recent years, yet it still offers a promising outlook for future research. As data continues to grow in complexity, diversity, and scale, there is a critical need to develop methods that are not only robust and adaptive but also efficient in handling such challenges. Below, we outline key directions for future research, emphasizing enhancements that could significantly improve the effectiveness and applicability of single-view and multi-view spectral clustering across diverse domains:

A. Finding Good Neighbors for Graph Construction

Future research should prioritize the development of advanced Good Neighbor strategies that identify the most representative and relevant neighbors for each data point. Ensuring the similarity graph accurately captures the underlying data structure is essential for effective clustering. This could involve investigating new criteria or metrics for neighbor selection that go beyond traditional distance-based methods. Additionally, incorporating domain-specific knowledge or machine learning techniques can further refine this selection process, enabling the construction of more reliable and context-aware similarity graphs.

B. Graph Construction Using Similarity and Dissimilarity Information

Incorporating both similarity and dissimilarity information into adaptive neighbor selection processes offers a promising direction for enhancing clustering performance. By dynamically adjusting neighbors based on both positive and negative relationships between data points, future methods could achieve greater robustness and accuracy. This approach may benefit from advancements in metric learning or semi-supervised learning, allowing for the guided construction of more meaningful and discriminative graphs. Exploring the interplay between similarity and dissimilarity in graph construction could yield more nuanced and flexible clustering outcomes.

C. High-Order Adaptive Neighbor Selection

High-order relationships, such as those represented by motifs or hypergraphs, provide richer insights into the structural characteristics of data compared to simple pairwise connections. Future research could focus on developing high-order adaptive neighbor selection methods that consider not only direct connections but also the broader relational context in which these connections occur. This may involve constructing higher-dimensional similarity matrices to capture intricate interactions among data points, resulting in more precise and interpretable clustering results. Such advancements could significantly enhance the ability of clustering methods to capture the true complexity of data structures.

D. Semi-Supervised Multi-View Spectral Clustering

The integration of semi-supervised techniques into spectral clustering holds significant potential for improving performance by leveraging limited labeled data. Future efforts could aim to develop semi-supervised multi-view spectral clustering methods that effectively combine labeled and unlabeled data across multiple views. This could include designing innovative regularization terms or loss functions that incorporate label information, as well as creating novel strategies to propagate labels through the spectral embedding process. Such methods could bridge the gap between supervised and unsupervised approaches, leading to more informed and accurate clustering results.

E. Multi-View Spectral Clustering Using Adaptive Hypergraphs

Hypergraphs represent an advanced framework for capturing complex multi-way relationships within data. Future research could explore adaptive hypergraph-based methods for multiview spectral clustering, where the hypergraph structure dynamically adjusts to the characteristics of the data. This may involve developing novel techniques for constructing hypergraphs that account for the varying importance of different views, or creating efficient algorithms to process these structures for clustering purposes. By harnessing the adaptability and expressiveness of hypergraphs, such methods could deliver more accurate and insightful clustering outcomes, especially in scenarios involving multi-view data.

VI. CONCLUSION

This paper has presented a systematic survey of spectral clustering methods, focusing on the fundamental role

of Graph Structure Learning (GSL) in improving clustering performance. We have examined diverse graph construction approaches, from pairwise and anchor-based methods to hypergraph techniques, analyzing both fixed and adaptive implementations. Our review encompasses single-view and multi-view frameworks, including their applications in one-step and two-step clustering processes, along with effective multi-view information fusion strategies. Through extensive analysis of various graph construction techniques and their impact on clustering accuracy, we highlight the importance of GSL in handling high-dimensional and large-scale datasets. By providing this comprehensive analysis of current methods and emerging challenges, we aim to guide future research in spectral clustering, particularly in developing efficient graph construction techniques and scalable solutions for large-scale data applications. This work contributes to the advancement of spectral clustering as an essential tool in modern data analysis and machine learning.

REFERENCES

[1] J. Xue, L. Xing, Y. Wang, X. Fan, L. Kong, Q. Zhang, F. Nie, and X. Li, "A comprehensive survey of fast graph clustering," *Vicinagearth*, vol. 1, no. 1, pp. 1–22, 2024.

[2] J. Xie, W. Kong, S. Xia, G. Wang, and X. Gao, "An efficient spectral clustering algorithm based on granular-ball," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 9, pp. 9743–9753, 2023.

[3] J. Singh and D. Singh, "A comprehensive review of clustering techniques in artificial intelligence for knowledge discovery: Taxonomy, challenges, applications and future prospects," *Advanced Engineering Informatics*, vol. 62, p. 102799, 2024.

[4] Q. Feng, C. P. Chen, and L. Liu, "A review of convex clustering from multiple perspectives: models, optimizations, statistical properties, applications, and connections," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[5] T. Benslama and R. Jallouli, "Clustering of social media data and marketing decisions," in *Digital Economy. Emerging Technologies and Business Innovation: 5th International Conference on Digital Economy, ICDEc 2020, Bucharest, Romania, June 11–13, 2020, Proceedings 5*. Springer, 2020, pp. 53–65.

[6] W. Liu, D. Gong, M. Tan, J. Q. Shi, Y. Yang, and A. G. Hauptmann, "Learning distilled graph for large-scale social network data clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 7, pp. 1393–1404, 2019.

[7] X. Liang, D. Wu, and D.-S. Huang, "Image co-segmentation via locally biased discriminative clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 11, pp. 2228–2233, 2019.

[8] D. J. Higham, G. Kalna, and M. Kibble, "Spectral clustering and its use in bioinformatics," *Journal of computational and applied mathematics*, vol. 204, no. 1, pp. 25–37, 2007.

[9] J. Li, H. Izakian, W. Pedrycz, and I. Jamal, "Clustering-based anomaly detection in multivariate time series data," *Applied Soft Computing*, vol. 100, p. 106919, 2021.

[10] D. Cai, X. He, and J. Han, "Document clustering using locality preserving indexing," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 12, pp. 1624–1637, 2005.

[11] J. Wang, C. Zhu, Y. Zhou, X. Zhu, Y. Wang, and W. Zhang, "From partition-based clustering to density-based clustering: Fast find clusters with diverse shapes and densities in spatial databases," *IEEE access*, vol. 6, pp. 1718–1729, 2017.

[12] J. Shao, X. He, C. Böhm, Q. Yang, and C. Plant, "Synchronization-inspired partitioning and hierarchical clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 4, pp. 893–905, 2012.

[13] L. Colomba, L. Cagliero, and P. Garza, "Density-based clustering by means of bridge point identification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 11, pp. 11 274–11 287, 2022.

[14] H. Deng and J. Han, "Probabilistic models for clustering," in *Data Clustering*. Chapman and Hall/CRC, 2018, pp. 61–86.

[15] F. Murtagh and P. Contreras, "Algorithms for hierarchical clustering: an overview, ii," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 7, no. 6, p. e1219, 2017.

[16] M. Unger and A. Tuzhilin, "Hierarchical latent context representation for context-aware recommendations," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 7, pp. 3322–3334, 2020.

[17] M. Hahsler, M. Piekenbrock, and D. Doran, "dbscan: Fast density-based clustering with r," *Journal of Statistical Software*, vol. 91, pp. 1–30, 2019.

[18] K. Berahmand, M. Mohammadi, F. Saberi-Movahed, Y. Li, and Y. Xu, "Graph regularized nonnegative matrix factorization for community detection in attributed networks," *IEEE Transactions on Network Science and Engineering*, vol. 10, no. 1, pp. 372–385, 2022.

[19] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, pp. 395–416, 2007.

[20] W. Liang, S. Zhou, J. Xiong, X. Liu, S. Wang, E. Zhu, Z. Cai, and X. Xu, "Multi-view spectral clustering with high-order optimal neighborhood laplacian matrix," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 7, pp. 3418–3430, 2020.

[21] M. Filippone, F. Camastra, F. Masulli, and S. Rovetta, "A survey of kernel and spectral methods for clustering," *Pattern recognition*, vol. 41, no. 1, pp. 176–190, 2008.

[22] Y. Ren, J. Pu, Z. Yang, J. Xu, G. Li, X. Pu, S. Y. Philip, and L. He, "Deep clustering: A comprehensive survey," *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[23] L. Hagen and A. B. Kahng, "New spectral methods for ratio cut partitioning and clustering," *IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems*, vol. 11, no. 9, pp. 1074–1085, 1992.

[24] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[25] Y. Yang, F. Shen, Z. Huang, H. T. Shen, and X. Li, "Discrete nonnegative spectral clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 9, pp. 1834–1845, 2017.

[26] D. Wu, F. Nie, J. Lu, R. Wang, and X. Li, "Effective clustering via structured graph learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 8, pp. 7909–7920, 2022.

[27] Q. Qiang, B. Zhang, F. Wang, and F. Nie, "Fast multi-view discrete clustering with anchor graphs," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 11, 2021, pp. 9360–9367.

[28] B. Yang, X. Zhang, Z. Lin, F. Nie, B. Chen, and F. Wang, "Efficient and robust multiview clustering with anchor graph regularization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 9, pp. 6200–6213, 2022.

[29] S. Zhang, S. Cui, and Z. Ding, "Hypergraph spectral clustering for point cloud segmentation," *IEEE Signal Processing Letters*, vol. 27, pp. 1655–1659, 2020.

[30] K. Ahn, K. Lee, and C. Suh, "Hypergraph spectral clustering in the weighted stochastic block model," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 5, pp. 959–974, 2018.

[31] C. Zhang, G. Zhu, B. Lian, M. Chen, H. Chen, and C. Wu, "Image segmentation based on multiscale fast spectral clustering," *Multimedia Tools and Applications*, vol. 80, pp. 24 969–24 994, 2021.

[32] R. Janani and S. Vijayarani, "Text document clustering using spectral clustering algorithm with particle swarm optimization," *Expert Systems with Applications*, vol. 134, pp. 192–200, 2019.

[33] R. Langone, C. Alzate, B. De Ketelaere, J. Vlasselaer, W. Meert, and J. A. Suykens, "Ls-svm based spectral clustering and regression for predicting maintenance of industrial machines," *Engineering Applications of Artificial Intelligence*, vol. 37, pp. 268–278, 2015.

[34] L. Ding, C. Li, D. Jin, and S. Ding, "Survey of spectral clustering based on graph theory," *Pattern Recognition*, p. 110366, 2024.

[35] Y. Pang, J. Xie, F. Nie, and X. Li, "Spectral clustering by joint spectral embedding and spectral rotation," *IEEE transactions on cybernetics*, vol. 50, no. 1, pp. 247–258, 2018.

[36] J. Huang, F. Nie, and H. Huang, "Spectral rotation versus k-means in spectral clustering," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 27, no. 1, 2013, pp. 431–437.

[37] Z. Kang, Z. Lin, X. Zhao, and W. Xu, "Structured graph learning for scalable subspace clustering: From single view to multiview," *IEEE Transactions on Cybernetics*, vol. 52, no. 9, pp. 8976–8986, 2021.

[38] Y. Zhu, W. Xu, J. Zhang, Q. Liu, S. Wu, and L. Wang, "Deep graph structure learning for robust representations: A survey," *arXiv preprint arXiv:2103.03036*, vol. 14, pp. 1–1, 2021.

[39] B. Li, Y.-R. Li, and X.-L. Zhang, "A survey on laplacian eigenmaps based manifold learning methods," *Neurocomputing*, vol. 335, pp. 336–351, 2019.

[40] Y. Chong, Y. Ding, Q. Yan, and S. Pan, "Graph-based semi-supervised learning: A review," *Neurocomputing*, vol. 408, pp. 216–230, 2020.

[41] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," *Advances in Neural Information Processing Systems*, vol. 14, 2001.

[42] F. Nie, X. Wang, and H. Huang, "Clustering and projected clustering with adaptive neighbors," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 977–986.

[43] X. Zhu, W. He, Y. Li, Y. Yang, S. Zhang, R. Hu, and Y. Zhu, "One-step spectral clustering via dynamically learning affinity matrix and subspace," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, 2017.

[44] F. Nie, X. Wang, M. Jordan, and H. Huang, "The constrained Laplacian rank algorithm for graph-based clustering," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, no. 1, 2016.

[45] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 11, pp. 2765–2781, 2013.

[46] F. Nie, J. Xue, W. Yu, and X. Li, "Fast clustering with anchor guidance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[47] F. Nie, W. Zhu, and X. Li, "Unsupervised large graph embedding based on balanced and hierarchical K-means," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 4, pp. 2008–2019, 2020.

[48] J. Wang, Z. Ma, F. Nie, and X. Li, "Efficient discrete clustering with anchor graph," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[49] D. Cai and X. Chen, "Large scale spectral clustering via landmark-based sparse representation," *IEEE Transactions on Cybernetics*, vol. 45, no. 8, pp. 1669–1680, 2015.

[50] J. Zhu, L. Tao, H. Yang, and F. Nie, "Unsupervised optimized bipartite graph embedding," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 3, pp. 3224–3238, 2021.

[51] X. Yang, M. Zhu, Y. Cai, Z. Wang, and F. Nie, "Fast spectral clustering with self-adapted bipartite graph learning," *Information Sciences*, vol. 644, p. 118810, 2023.

[52] D. Zhou, J. Huang, and B. Schölkopf, "Learning with hypergraphs: Clustering, classification, and embedding," *Advances in Neural Information Processing Systems*, vol. 19, 2006.

[53] Y. Gao, Z. Zhang, H. Lin, X. Zhao, S. Du, and C. Zou, "Hypergraph learning: Methods and practices," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 5, pp. 2548–2566, 2020.

[54] A. Antelmi, G. Cordasco, M. Polato, V. Scarano, C. Spagnuolo, and D. Yang, "A survey on hypergraph representation learning," *ACM Computing Surveys*, vol. 56, no. 1, pp. 1–38, 2023.

[55] X. Zhu, S. Zhang, Y. Zhu, P. Zhu, and Y. Gao, "Unsupervised spectral feature selection with dynamic hyper-graph learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 6, pp. 3016–3028, 2020.

[56] Y. Huang, Q. Liu, and D. Metaxas, "Video object segmentation by hypergraph cut," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 1738–1745.

[57] Y. Gao, M. Wang, D. Tao, R. Ji, and Q. Dai, "3-D object retrieval and recognition with hypergraph analysis," *IEEE Transactions on Image Processing*, vol. 21, no. 9, pp. 4290–4303, 2012.

[58] M. Wang, X. Liu, and X. Wu, "Visual classification by ℓ_1-hypergraph modeling," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 9, pp. 2564–2574, 2015.

[59] T. Jin, Z. Yu, Y. Gao, S. Gao, X. Sun, and C. Li, "Robust ℓ_2-hypergraph and its applications," *Information Sciences*, vol. 501, pp. 708–723, 2019.

[60] Q. Liu, Y. Sun, C. Wang, T. Liu, and D. Tao, "Elastic net hypergraph learning for image clustering and semi-supervised classification," *IEEE Transactions on Image Processing*, vol. 26, no. 1, pp. 452–463, 2016.

[61] X. Zhu, Y. Zhu, S. Zhang, R. Hu, and W. He, "Adaptive hypergraph learning for unsupervised feature selection," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 2017, pp. 3581–3587.

[62] B. Ghojogh, M. Crowley, F. Karray, and A. Ghodsi, *Elements of dimensionality reduction and manifold learning*. Springer, 2023.

[63] Y. Yun, J. Li, Q. Gao, M. Yang, and X. Gao, "Low-rank discrete multi-view spectral clustering," *Neural Networks*, vol. 166, pp. 137–147, 2023.

[64] Z. Hu, F. Nie, R. Wang, and X. Li, "Multi-view spectral clustering via integrating nonnegative embedding and spectral embedding," *Information Fusion*, vol. 55, pp. 251–259, 2020.

[65] D. Kuang, S. Yun, and H. Park, "SymNMF: nonnegative low-rank approximation of a similarity matrix for graph clustering," *Journal of Global Optimization*, vol. 62, no. 3, pp. 545–574, 2015.

[66] M. Luo, C. Yan, Q. Zheng, X. Chang, L. Chen, and F. Nie, "Discrete multi-graph clustering," *IEEE Transactions on Image Processing*, vol. 28, no. 9, pp. 4701–4712, 2019.

[67] J. Han, K. Xiong, and F. Nie, "Orthogonal and nonnegative graph reconstruction for large scale clustering," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 2017, pp. 1809–1815.

[68] J.-H. Kim, J.-H. Choi, Y.-H. Park, C. K.-S. Leung, and A. Nasridinov, "Knn-sc: novel spectral clustering algorithm using k-nearest neighbors," *IEEE Access*, vol. 9, pp. 152 616–152 627, 2021.

[69] F. Nie, D. Wu, R. Wang, and X. Li, "Self-weighted clustering with adaptive neighbors," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 9, pp. 3428–3441, 2020.

[70] J. Wang, Z. Ma, F. Nie, and X. Li, "Entropy regularization for unsupervised clustering with adaptive neighbors," *Pattern Recognition*, vol. 125, p. 108517, 2022.

[71] X. Zhu, Y. Zhu, and W. Zheng, "Spectral rotation for deep one-step clustering," *Pattern Recognition*, vol. 105, p. 107175, 2020.

[72] Y. Zhao and X. Li, "Spectral clustering with adaptive neighbors for deep learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 4, pp. 2068–2078, 2021.

[73] J. Zhou, C. Gao, X. Wang, Z. Lai, J. Wan, and X. Yue, "Typicality-aware adaptive similarity matrix for unsupervised learning," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[74] Z. Li, F. Nie, X. Chang, L. Nie, H. Zhang, and Y. Yang, "Rank-constrained spectral clustering with flexible embedding," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 12, pp. 6073–6082, 2018.

[75] X. Li, B. Kao, Z. Ren, and D. Yin, "Spectral clustering in heterogeneous information networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 4221–4228.

[76] Y. Duan, F. Nie, R. Wang, and X. Li, "New approach for learning structured graph with laplacian rank constraint," *Neurocomputing*, p. 128065, 2024.

[77] Q. Wang, Z. Qin, F. Nie, and X. Li, "Spectral embedded adaptive neighbors clustering," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 4, pp. 1265–1271, 2018.

[78] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 171–184, 2012.

[79] C.-Y. Lu, H. Min, Z.-Q. Zhao, L. Zhu, D.-S. Huang, and S. Yan, "Robust and efficient subspace segmentation via least squares regression," in *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VII 12*. Springer, 2012, pp. 347–360.

[80] H. Hu, Z. Lin, J. Feng, and J. Zhou, "Smooth representation clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 3834–3841.

[81] J. Liu, Y. Chen, J. Zhang, and Z. Xu, "Enhancing low-rank subspace clustering by manifold regularization," *IEEE Transactions on Image Processing*, vol. 23, no. 9, pp. 4022–4030, 2014.

[82] M. Yin, J. Gao, Z. Lin, Q. Shi, and Y. Guo, "Dual graph regularized latent low-rank representation for subspace clustering," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 4918–4933, 2015.

[83] S. Du, Y. Ma, and Y. Ma, "Graph regularized compact low rank representation for subspace clustering," *Knowledge-Based Systems*, vol. 118, pp. 56–69, 2017.

[84] P. Ji, T. Zhang, H. Li, M. Salzmann, and I. Reid, "Deep subspace clustering networks," *Advances in neural information processing systems*, vol. 30, 2017.

[85] X. Peng, J. Feng, S. Xiao, W.-Y. Yau, J. T. Zhou, and S. Yang, "Structured autoencoders for subspace clustering," *IEEE Transactions on Image Processing*, vol. 27, no. 10, pp. 5076–5086, 2018.

[86] J. Wen, X. Fang, Y. Xu, C. Tian, and L. Fei, "Low-rank representation with adaptive graph regularization," *Neural Networks*, vol. 108, pp. 83–96, 2018.

[87] X. Zhu, S. Zhang, Y. Li, J. Zhang, L. Yang, and Y. Fang, "Low-rank sparse subspace for spectral clustering," *IEEE Transactions on knowledge and data engineering*, vol. 31, no. 8, pp. 1532–1543, 2018.

[88] T. Tong, J. Gan, G. Wen, and Y. Li, "One-step spectral clustering based on self-paced learning," *Pattern Recognition Letters*, vol. 135, pp. 8–14, 2020.

[89] W. Dong, X.-J. Wu, and J. Kittler, "Subspace clustering via joint $\ell_{1,2}$ and $\ell_{2,1}$ norms," *Information Sciences*, vol. 612, pp. 675–686, 2022.

[90] W. Zhu, F. Nie, and X. Li, "Fast spectral clustering with efficient large graph construction," in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 2492–2496.

[91] R. Wang, F. Nie, and W. Yu, "Fast spectral clustering with anchor graph for large hyperspectral images," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 11, pp. 2003–2007, 2017.

[92] R. Wang, F. Nie, Z. Wang, F. He, and X. Li, "Scalable graph-based clustering with nonnegative relaxation for large hyperspectral image," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 10, pp. 7352–7364, 2019.

[93] B. Yang, J. Wu, A. Sun, N. Gao, and X. Zhang, "Robust landmark graph-based clustering for high-dimensional data," *Neurocomputing*, vol. 496, pp. 72–84, 2022.

[94] J. Chen, J. Zhu, S. Xie, H. Yang, and F. Nie, "FGC_SS: Fast graph clustering method by joint spectral embedding and improved spectral rotation," *Information Sciences*, vol. 613, pp. 853–870, 2022.

[95] G. Yang, S. Deng, X. Chen, C. Chen, Y. Yang, Z. Gong, and Z. Hao, "Reskm: a general framework to accelerate large-scale spectral clustering," *Pattern Recognition*, vol. 137, p. 109275, 2023.

[96] Z. Wang, X. Dai, P. Zhu, R. Wang, X. Li, and F. Nie, "Fast optimization of spectral embedding and improved spectral rotation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 2, pp. 1515–1527, 2021.

[97] C. Gao, W. Chen, F. Nie, W. Yu, and Z. Wang, "Spectral clustering with linear embedding: A discrete clustering method for large-scale data," *Pattern Recognition*, vol. 151, p. 110396, 2024.

[98] S. Huang, A. Elgammal, and D. Yang, "On the effect of hyperedge weights on hypergraph learning," *Image and Vision Computing*, vol. 57, pp. 89–101, 2017.

[99] D. Ghoshdastidar and A. Dukkipati, "A provable generalized tensor spectral method for uniform hypergraph partitioning," in *International Conference on Machine Learning*. PMLR, 2015, pp. 400–409.

[100] L. Pu and B. Faltings, "Hypergraph learning with hyperedge expansion," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2012, pp. 410–425.

[101] S. Saito, D. Mandic, and H. Suzuki, "Hypergraph p-laplacian: A differential geometry view," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[102] C. Yang, R. Wang, S. Yao, and T. Abdelzaher, "Hypergraph learning with line expansion," *arXiv preprint arXiv:2005.04843*, 2020.

[103] Y. Huang, Q. Liu, F. Lv, Y. Gong, and D. N. Metaxas, "Unsupervised image categorization by hypergraph partition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 6, pp. 1266–1273, 2011.

[104] X. Li, W. Hu, C. Shen, A. Dick, and Z. Zhang, "Context-aware hypergraph construction for robust spectral clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 10, pp. 2588–2597, 2013.

[105] Y. Yang, S. Deng, J. Lu, Y. Li, Z. Gong, Z. Hao *et al.*, "Graphlshc: towards large scale spectral hypergraph clustering," *Information Sciences*, vol. 544, pp. 117–134, 2021.

[106] S. Huang, D. Yang, B. Liu, and X. Zhang, "Regression-based hypergraph learning for image clustering and classification arxiv," in *Comput. Vis. Pattern Recognit.*, 2016.

[107] T. Jin, R. Ji, Y. Gao, X. Sun, X. Zhao, and D. Tao, "Correntropy-induced robust low-rank hypergraph," *IEEE Transactions on Image Processing*, vol. 28, no. 6, pp. 2755–2769, 2018.

[108] T. Jin, L. Cao, F. Jie, and R. Ji, "Link-aware semi-supervised hypergraph," *Information Sciences*, vol. 507, pp. 339–355, 2020.

[109] L. Li, Z. Li, M. Wang, T. Jin, and J. Liu, "Hessian non-negative hypergraph," in *International Conference on Intelligent Computing*. Springer, 2023, pp. 691–704.

[110] U. Fang, M. Li, J. Li, L. Gao, T. Jia, and Y. Zhang, "A comprehensive survey on multi-view clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 12, pp. 12350–12368, 2023.

[111] Y. Li, M. Yang, and Z. Zhang, "A survey of multi-view representation learning," *IEEE transactions on knowledge and data engineering*, vol. 31, no. 10, pp. 1863–1883, 2018.

[112] L. Zhou, G. Du, K. Lü, L. Wang, and J. Du, "A survey and an empirical evaluation of multi-view clustering approaches," *ACM Computing Surveys*, vol. 56, no. 7, pp. 1–38, 2024.

[113] Z. Hu, F. Nie, W. Chang, S. Hao, R. Wang, and X. Li, "Multi-view spectral clustering via sparse graph learning," *Neurocomputing*, vol. 384, pp. 1–10, 2020.

[114] F. Nie, J. Li, X. Li *et al.*, "Parameter-free auto-weighted multiple graph learning: A framework for multiview clustering and semi-supervised classification." in *IJCAI*, vol. 9, 2016, pp. 1881–1887.

[115] ——, "Self-weighted multiview clustering with multiple graphs." in *IJCAI*, 2017, pp. 2564–2570.

[116] F. Nie, G. Cai, J. Li, and X. Li, "Auto-weighted multi-view learning for image clustering and semi-supervised classification," *IEEE Transactions on Image Processing*, vol. 27, no. 3, pp. 1501–1511, 2017.

[117] X. Liu, X. Zhu, M. Li, L. Wang, C. Tang, J. Yin, D. Shen, H. Wang, and W. Gao, "Late fusion incomplete multi-view clustering," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 10, pp. 2410–2423, 2018.

[118] J. Tan, Y. Shi, Z. Yang, C. Wen, and L. Lin, "Unsupervised multi-view clustering by squeezing hybrid knowledge from cross view and each view," *IEEE Transactions on Multimedia*, vol. 23, pp. 2943–2956, 2020.

[119] M. He, S.-J. Horng, P. Fan, R.-S. Run, R.-J. Chen, J.-L. Lai, M. K. Khan, and K. O. Sentosa, "Performance evaluation of score level fusion in multimodal biometric systems," *Pattern Recognition*, vol. 43, no. 5, pp. 1789–1800, 2010.

[120] R. Bai, R. Huang, Y. Qin, Y. Chen, and Y. Xu, "A structural consensus representation learning framework for multi-view clustering," *Knowledge-Based Systems*, vol. 283, p. 111132, 2024.

[121] Y. Hao, X.-Y. Jing, R. Chen, and W. Liu, "Learning enhanced specific representations for multi-view feature learning," *Knowledge-Based Systems*, vol. 272, p. 110590, 2023.

[122] L. Zong, X. Zhang, X. Liu, and H. Yu, "Weighted multi-view spectral clustering based on spectral perturbation," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[123] S. Shi, F. Nie, R. Wang, and X. Li, "Auto-weighted multi-view clustering via spectral embedding," *Neurocomputing*, vol. 399, pp. 369–379, 2020.

[124] H. Wang, Y. Yang, B. Liu, and H. Fujita, "A study of graph-based system for multi-view clustering," *Knowledge-Based Systems*, vol. 163, pp. 1009–1019, 2019.

[125] H. Wang, Y. Yang, and B. Liu, "Gmc: Graph-based multi-view clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 6, pp. 1116–1129, 2019.

[126] S. El Hajjar, F. Dornaika, F. Abdallah, and N. Barrena, "Consensus graph and spectral representation for one-step multi-view kernel based clustering," *Knowledge-Based Systems*, vol. 241, p. 108250, 2022.

[127] G. Zhong, T. Shu, G. Huang, and X. Yan, "Multi-view spectral clustering by simultaneous consensus graph learning and discretization," *Knowledge-Based Systems*, vol. 235, p. 107632, 2022.

[128] J. Chen, H. Mao, D. Peng, C. Zhang, and X. Peng, "Multiview clustering by consensus spectral rotation fusion," *IEEE Transactions on Image Processing*, 2023.

[129] C. Tang, M. Wang, and K. Sun, "One-step multiview clustering via adaptive graph learning and spectral rotation," *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[130] X. Cao, C. Zhang, H. Fu, S. Liu, and H. Zhang, "Diversity-induced multi-view subspace clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 586–594.

[131] Y. Wang, W. Zhang, L. Wu, X. Lin, M. Fang, and S. Pan, "Iterative views agreement: An iterative low-rank based structured optimization method to multi-view spectral clustering," *arXiv preprint arXiv:1608.05560*, 2016.

[132] X. Wang, X. Guo, Z. Lei, C. Zhang, and S. Z. Li, "Exclusivity-consistency regularized multi-view subspace clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 923–931.

[133] M. Brbić and I. Kopriva, "Multi-view low-rank sparse subspace clustering," *Pattern recognition*, vol. 73, pp. 247–258, 2018.

[134] S. Luo, C. Zhang, W. Zhang, and X. Cao, "Consistent and specific multi-view subspace clustering," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[135] R. Li, C. Zhang, Q. Hu, P. Zhu, and Z. Wang, "Flexible multi-view representation learning for subspace clustering." in *IJCAI*, vol. 2019, 2019, pp. 2916–2922.

[136] Z. Kang, G. Shi, S. Huang, W. Chen, X. Pu, J. T. Zhou, and Z. Xu, "Multi-graph fusion for multi-view spectral clustering," *Knowledge-Based Systems*, vol. 189, p. 105102, 2020.

[137] W. Hao, S. Pang, and Z. Chen, "Multi-view spectral clustering via common structure maximization of local and global representations," *Neural Networks*, vol. 143, pp. 595–606, 2021.

[138] Q. Wang, J. Cheng, Q. Gao, G. Zhao, and L. Jiao, "Deep multi-view subspace clustering with unified and discriminative learning," *IEEE Transactions on Multimedia*, vol. 23, pp. 3483–3493, 2020.

[139] P. Zhang, X. Liu, J. Xiong, S. Zhou, W. Zhao, E. Zhu, and Z. Cai, "Consensus one-step multi-view subspace clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 10, pp. 4676–4689, 2020.

[140] M.-S. Chen, L. Huang, C.-D. Wang, D. Huang, and S. Y. Philip, "Multiview subspace clustering with grouping effect," *IEEE Transactions on Cybernetics*, vol. 52, no. 8, pp. 7655–7668, 2020.

[141] Z. Lin, Z. Kang, L. Zhang, and L. Tian, "Multi-view attributed graph clustering," *IEEE Transactions on knowledge and data engineering*, vol. 35, no. 2, pp. 1872–1880, 2021.

[142] Y. Li, F. Nie, H. Huang, and J. Huang, "Large-scale multi-view spectral clustering via bipartite graph," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 29, no. 1, 2015.

[143] X. Li, H. Zhang, R. Wang, and F. Nie, "Multiview clustering: A scalable and parameter-free bipartite graph fusion method," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 1, pp. 330–344, 2020.

[144] L. Li and H. He, "Bipartite graph based multi-view clustering," *IEEE transactions on knowledge and data engineering*, vol. 34, no. 7, pp. 3111–3125, 2020.

[145] B. Yang, X. Zhang, F. Nie, and F. Wang, "Fast multiview clustering with spectral embedding," *IEEE Transactions on Image Processing*, vol. 31, pp. 3884–3895, 2022.

[146] B. Yang, X. Zhang, J. Wu, F. Nie, Z. Lin, F. Wang, and B. Chen, "Fast multiview anchor-graph clustering," *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[147] W. Hao, S. Pang, J. Zhu, and Y. Li, "Self-weighting and hypergraph regularization for multi-view spectral clustering," *IEEE Signal Processing Letters*, vol. 27, pp. 1325–1329, 2020.

[148] G. Yang, Q. Li, Y. Yun, Y. Lei, and J. You, "Hypergraph learning-based semi-supervised multi-view spectral clustering," *Electronics*, vol. 12, no. 19, p. 4083, 2023.

[149] J. Chen, H. Xu, J. Xue, Q. Gao, C. Deng, and Z. Lv, "Incomplete multi-view clustering based on hypergraph," *Information Fusion*, p. 102804, 2024.

Yuefeng Li is currently a professor and the HDR Director with the School of Computer Science, Queensland University of Technology (QUT), Australia. His work has received 7,146 citations, with an overall h-index of 41 and an i10 index of 138. He has published 12 articles with more than 100 citations and 5 articles with more than 200 citations. Professor Yuefeng Li carries out research on data mining, text analysis, and AI-based Data Analysis. He has a lot of experience in developing new data mining and machine learning algorithms in all data types (structured and unstructured). He is recognized internationally in Knowledge & Data Engineering and Natural Language Processing. He has demonstrable experience in leading large-scale research projects and has achieved many established research outcomes that have been published and highly cited in top data mining journals and conferences, such as the TKDE, ACM Trans. Knowl. Discov. Data, Data Min. Knowl. Discov., IEEE Trans. Netw. Sci. Eng., Information Sciences, Artificial Intelligence Review, KDD, CIKM, WWW, ICDM, and PAKDD. He is an Editor-in-Chief of Web Intelligence journal.

Kamal Berahmand received his Ph.D. in 2024 from the School of Computer Science, Queensland University of Technology (QUT), Australia, where he is now a researcher. His current research interests include machine learning, representation learning, graph neural networks, and graph learning.

Farid Saberi-Movahed earned his Ph.D. in Applied Mathematics, specializing in Optimization and Numerical Linear Algebra, from Shahid Bahonar University of Kerman, Iran. Currently, he serves as an Associate Professor in the Department of Applied Mathematics, Faculty of Sciences and Modern Technologies, at the Graduate University of Advanced Technology, Kerman, Iran. His research interests focus on numerical linear algebra methods, with an emphasis on their applications in Data Mining and Machine Learning.

Razieh Sheikhpour received her Ph.D. in Computer Engineering from Yazd University, Yazd, Iran, in 2017. Currently, she is an Associate Professor at the department of Computer Engineering at Ardakan University, Ardakan, Iran. Her research interests include machine learning, semi-supervised feature selection and bioinformatics.

Mahdi Jalili (Senior Member, IEEE) received the B.S. degree in electrical engineering from Tehran Polytechnic, Tehran, Iran, in 2001, the M.S. degree in electrical engineering from the University of Tehran, Tehran, in 2004, and the Ph.D. degree in synchronization in dynamical networks from the Swiss Federal Institute of Technology Lausanne, Lausanne, Switzerland. He then joined the Sharif University of Technology as an Assistant Professor. He is currently a Senior Lecturer with the School of Engineering, RMIT University, Melbourne, VIC, Australia. His research interests include network science, dynamical systems, social networks analysis and mining, and data analytics. Dr. Jalili was a recipient of the Australian Research Council DECRA Fellowship and the RMIT Vice-Chancellor Research Fellowship. He is an Associate Editor of the IEEE Canadian Journal of Electrical and Computer Engineering and an Editorial Board Member of the Mathematical Problems in Engineering and Complex Adaptive Systems Modeling.