Adaptive Unified Framework with Global Anchor Graph for Large-scale Multi-view Clustering

Lin Shi, Wangjie Chen, Yi Liu, Lihua Zhuang, and Guangqi Jiang(✉)

School of Computer Science and Artificial Intelligence, Changzhou University, Changzhou, 213164, China
guangqijiang@cczu.edu.cn

Abstract. Multi-view clustering faces serious challenges in reducing computational and memory demands for large-scale datasets while effectively extracting structural information from multi-view data. Most existing methods address algorithmic complexity by introducing anchors, typically through a two-stage process involving anchor sampling and subsequent bipartite graph construction. However, the quality of anchor selection directly affects the performance of the bipartite graph, this two-stage mechanism lacks mutual optimization, thereby negatively impacting clustering performance. To address these issues, we propose the Adaptive Unified Framework with Global Anchor Graph for Large-scale Multi-view Clustering (AUF-LMC). Different from the traditional sample-based anchor selection mechanism, AUF-LMC adaptively learns the underlying anchors across multiple views and builds global bipartite graph on this basis, so that these two processes can be linked to each other to promote optimization and improve clustering performance. Furthermore, we unify all processes within a single framework and apply appropriate constraints to the bipartite graph. Experimental evaluations demonstrate that our method delivers superior clustering performance and efficiency, characterized by fast convergence and robustness on standard datasets.

Keywords: Large-scale Multi-view Clustering · Bipartite Graph Learning · Learning-based Anchor Selection.

1 Introduction

With the rapid innovation of information technology and the explosion of data volume, the amount of data collected from various fields and perspectives is increasing, such as a 3D object can be described by different views or various devices[4],[20]. It is the complexity of such data that brings the majority of researchers to think deeply about the multi-view learning method. Among these methods, Multi-view Clustering (MVC)[8],[16],[7],[9] serves as a crucial technique that leverages datasets from multiple views to enhance clustering performance. The complementary information provided by different views is considered a key factor in improving cluster performance because it reflects the characteristics of the same data from various aspects. However, existing multi-view clustering

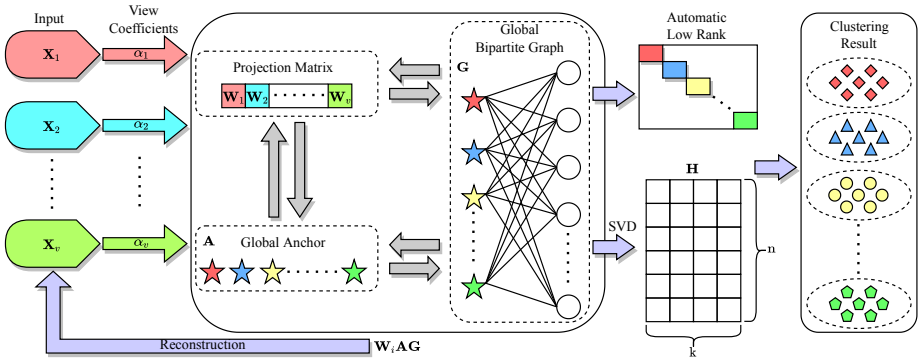

Fig. 1. The framework of AUF-LMC.

algorithms face serious challenges when processing large-scale datasets, especially in terms of computational costs and storage requirements.

Traditional Multi-view Subspace Clustering (MVSC) methods attempt to reveal the intrinsic structure of data by exploring a shared or a set of related subspaces in each view[15],[13],[23]. These methods typically adopt a self-representation model, representing each data point as a linear combination of other points within the same view. Although these approaches have achieved notable success in multi-view clustering tasks, the self-representation framework's computational complexity and memory requirements rapidly increase as the dataset size expands, severely limiting their scalability and practicality.

In recent years, to address these issues, some strategies[13],[12],[26],[24] based on anchors have been applied to multi-view clustering. These methods select a small number of representative data points as anchor points or landmarks to effectively approximate the original data and reduce complexity. Additionally, there are studies proposing the use of bipartite graphs[13],[3],[31],[30] to simplify the multi-view clustering problem by connecting data points and anchor points, effectively reducing the problem's dimensionality and computational cost. However, these methods often rely on heuristic anchor selection strategies, such as k-means or random sampling[2],[25], which may lead to suboptimal anchor selection, thus adversely affecting the clustering performance.

To address these challenges, this paper proposes a novel multi-view clustering method termed Adaptive Unified Framework with Global Anchor Graph for Large-scale Multi-view Clustering(AUF-LMC) that employs a learning-based anchor selection mechanism to automatically determine the optimal set of anchors, more accurately capturing the data underlying structure. The framework of AUF-LMC is shown in Fig. 1. We start by collecting data from multiple views as input, and each view is assigned a learnable view coefficient. The main part of the framework is the mutual promotion learning of projection matrix $\mathbf{W}_i$, global anchor matrix $\mathbf{A}$ and global bipartite graph matrix $\mathbf{G}$, and the reconstruction error measurement of input data is carried out through the learning of these

three. Then SVD is performed on the basis of the best global bipartite graph $\mathbf{G}$ to obtain spectral embedding $\mathbf{H}$ and clustering results. It should be noted that the learned global bipartite graph have low-rank attributes, which not only retain the main features of the original data, but also reduce the storage space and computational complexity of the data. By adaptively learning the underlying anchor graph across multiple views, AUF-LMC can effectively handle large-scale datasets and ensure consistency and complementarity among different views by optimizing a unified objective function. Additionally, the optimization strategy and computational complexity of AUF-LMC are discussed in detail and the results of comprehensive testing across a variety of benchmark datasets demonstrate enhanced clustering outcomes and improved efficiency over current multi-view clustering techniques.

Broadly, the significant contributions of this paper are as follows:

- **Innovative Learning-Based Anchor Selection Mechanism:** We introduce a novel multi-view clustering method that employs a learning-based anchor selection strategy to automatically identify and utilize the data underlying structure for optimal anchor set determination.
- **Unified Multi-View Clustering Framework:** We designed a adaptive unified framework that combines global anchor selection, bipartite graph construction and clustering processes to promote consistency and complementarity among different views, thereby improving clustering performance and stability.
- **Efficient Four-Step Alternating Optimization Algorithm:** Our method introduces an efficient four-step alternating optimization algorithm. By iteratively solving for one variable at a time, this algorithm not only ensures convergence to an optimal solution but also enhances the method's efficiency.

2 Related Works

In this section, we initially review the principles of multi-view subspace clustering and subsequently provide an overview of earlier methodologies for multi-view clustering that utilize bipartite graphs.

2.1 Multi-view Subspace Clustering

MVSC is based on the assumption that data from multiple sources can provide complementary information, and if they can be combined synergistically, they can provide a richer representation for clustering. Among the numerous MVSC research methods, one popular approach is to utilize self-representative techniques where data points in each view are expressed as a linear combination of other points within the same view[15],[13],[18],[19]. This self-expressiveness can be mathematically formulated as an optimization problem where the goal is to minimize the reconstruction error subject to certain constraints.

Given multi-view data $\{\mathbf{X}_i\}_{i=1}^v$ where $\mathbf{X}_i \in \mathbb{R}^{d_i \times n}$ and d_i representing the dimension of the i-th view and n denoting the number of data samples, the objective is to learn a set of self-expressive matrices $\{\mathbf{S}_i\}_{i=1}^v$ that capture the subspace structure in each view. The global structure is then inferred by integrating these self-expressive representations. This integration is usually done through a joint optimization framework, which can be expressed as:

$$\min_{\mathbf{S}_i} \sum_{i=1}^{v} \alpha_i \|\mathbf{X}_i - \mathbf{X}_i\mathbf{S}_i\|_{\mathbf{F}}^2 + \lambda \sum_{i=1}^{v} \|\mathbf{S}_i\|_*, \tag{1}$$

where α_i is a view-specific weight and λ is a regularization parameter. $\|\cdot\|_{\mathbf{F}}^2$ is the square of the Frobenius norm, which measures the reconstruction error. $\|\cdot\|_*$ represents the nuclear norm (or trace norm), allowing $\mathbf{S}_i$ to better capture the low-dimensional subspace structure of the data..

The consensus across multiple views can be achieved by aligning the subspaces spanned by the self-expressive matrices. This leads to a joint spectral embedding $\mathbf{H} \in \mathbb{R}^{n \times k}$ obtained by solving the optimization problem:

$$\max_{\mathbf{H}} \operatorname{Tr}\left(\mathbf{H}^\top \left(\frac{\mathbf{S} + \mathbf{S}^\top}{2}\right)\mathbf{H}\right), \quad \text{s.t. } \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k, \tag{2}$$

where $\operatorname{Tr}(\cdot)$ represents the trace function which is the sum of the diagonal elements of the matrix, $\mathbf{S}$ is the concatenated self-expressive matrix from all views, $\mathbf{H}^\top\mathbf{H} = \mathbf{I}_k$ is an orthogonality constraint ensuring that $\mathbf{H}$ is orthogonal, k is the number of categories. Spectral clustering is then applied to the spectral embedding $\mathbf{H}$, which serves as the input to algorithms like k-means for final cluster assignment.

The computational complexity of solving this optimization problem typically scales with the number of data points and the dimensionality of the embedding space, which necessitates efficient algorithms capable of handling large-scale data. Recent advancements in this field have focused on anchor-based methods that construct anchor graphs to significantly reduce both computational time and storage requirements, paving the way for scalable MVSC solutions.

2.2 Bipartite Graph for Multi-view Clustering

In the field of MVC, the challenge of processing large-scale datasets necessitates innovative strategies to reduce computational demands while retaining the richness of multi-view information. A bipartite graph forms a fundamental construct in this regard, with its two disjoint sets of nodes—typically, one representing the data instances and the other embodying the multiple views of the dataset.

In a multi-view dataset with n samples across v views and k clusters, each view features a matrix $\mathbf{X}_i \in \mathbb{R}^{d_i \times n}$. A multi-view bipartite graph is articulated as $\mathbf{G} = (\{\mathbf{X}_i\}_{i=1}^v, \Upsilon, \{\mathbf{B}_i\}_{i=1}^v)$, where $\Upsilon = \{\mathbf{Y}_1, \mathbf{Y}_2, ..., \mathbf{Y}_v\}$ denotes the set of anchor points. Here, $\mathbf{Y}_i \in \mathbb{R}^{d_i \times k}$ represents the features of k anchor points in the i-th view, and $\mathbf{B}_i \in \mathbb{R}^{k \times n}$ is the affinity matrix for the i-th view. This structure

elegantly captures the essence of multi-view relationships by bridging features and samples via anchor points, thereby enhancing clustering by focusing on the most representative connections.

The utilization of bipartite graphs in multi-view clustering is achieved by mapping the relationships between data points and features across different views onto the graph. This allows for the exploitation of inter-view connections, which are crucial for discerning the underlying data structure.

The primary advantage of employing a bipartite graph lies in its ability to reduce the size of the problem space. Instead of handling a full similarity matrix of dimension $\mathbb{R}^{n\times n}$, the problem is distilled to dealing with $\mathbb{R}^{k\times n}$, where k is typically much smaller than n, thus lowering both the storage requirements and the computational complexity. By sampling a smaller proportion of representative landmarks, the bipartite graph approach not only captures the essential relationships within the data but also significantly speeds up the clustering process, a clear benefit for large-scale data analysis where efficiency is paramount.

3 The Proposed Method

In this section, we delve into AUF-LMC in greater detail. Firstly, we elucidate the underlying motivation behind AUF-LMC to provide a comprehensive introduction of the model. Subsequently, we elaborate on the optimization methodology employed by the model with meticulous attention to detail. Lastly, we analyze the intricacies of its time complexity of AUF-LMC.

3.1 Problem Formulation

In the field of multi-view subspace clustering, popular methods rely on the principle of self-representation, representing each data point in the global context by considering the entire data population. Although this approach delves into the full range of relationships between data points, it comes with a significant computational and storage burden. In addition, many studies reduce complexity by introducing anchors, but traditional anchor-based subspace methods predominantly hinge on heuristic anchor selection strategies, such as k-means or random sampling, which are separated from the subsequent construction of bipartitic graphs and thus degrade clustering performance.

To address these challenges, this papser proposes a learning-based anchor strategy that automatically learns the optimal set of anchors and abandons reliance on heuristics to capture the significant features of the data and reflect the underlying structure of the data. Through this method, we combine anchor learning and bipartite graph learning closely to promote each other's optimization and improve the clustering performance.

The innovation of AUF-LMC lies not only in its anchor selection mechanism but also in the overarching optimization framework it introduces. The framework is equipped with an additional regularization term and some suitable constraints,

vital for mitigating model complexity and warding off overfitting. As a result, the objective function is articulated as:

$$\begin{aligned} &\min_{\boldsymbol{\alpha},\mathbf{W}_i,\mathbf{A},\mathbf{G}} \sum_{i=1}^{v} {\alpha_i}^2 \|\mathbf{X}_i - \mathbf{W}_i\mathbf{A}\mathbf{G}\|_{\mathbf{F}}^2 + \lambda\|\mathbf{G}\|_{\mathbf{F}}^2, \\ &\text{s.t. } \boldsymbol{\alpha}^\top\mathbf{1} = 1,\ \mathbf{W}_i^\top\mathbf{W}_i = \mathbf{I}_k, \mathbf{A}^\top\mathbf{A} = \mathbf{I}_k, \mathbf{G} \geq 0, \mathbf{G}^\top\mathbf{1} = \mathbf{1}. \end{aligned} \tag{3}$$

In Eq. (3), $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, ..., \alpha_v]^\top$ is the vector of all the view coefficients, $\mathbf{X}_i \in \mathbb{R}^{d_i \times n}$ denotes the data matrix for the i-th view, and $\mathbf{W}_i \in \mathbb{R}^{d_i \times k}$ is its corresponding projection matrix. The matrix $\mathbf{A} \in \mathbb{R}^{k \times k}$ is the global anchor matrix, and $\mathbf{G} \in \mathbb{R}^{k \times n}$ represents the global bipartite graph. The last term serves as the regularization component, imposing a penalty on the complexity of the model to mitigate overfitting and promote generalization.

The construction of the anchor graph $\mathbf{G}$ sets the stage for our spectral embedding process. To this end, we utilize the Gram matrix $\mathbf{G}^\top\Delta^{-1}\mathbf{G}$, where Δ is a diagonal matrix with entries $\Delta_{ii} = \sum_{j=1}^{n} \mathbf{G}_{ji}$, equating to the sum of each row in $\mathbf{G}$. The matrix Δ essentially normalizes $\mathbf{G}$, allowing us to derive the spectral embedding $\mathbf{H}$ in a manner that is more computationally efficient than traditional eigen-decomposition approaches.

Theorem 1. *The right singular vectors of* $\mathbf{G}$*, which are contained in the matrix* $\mathbf{V}$ *from the singular value decomposition* $\mathbf{G} = \mathbf{U}\Sigma\mathbf{V}^\top$*, are identical to the eigenvectors of* $\mathbf{G}^\top\Delta^{-1}\mathbf{G}$*.*

Theorem 1 indicates that the spectral embedding $\mathbf{H}$ can be efficiently calculated by an SVD on $\mathbf{G}$, necessitating only $O(nk^2)$ operations, a substantial improvement from the classical $O(n^3)$. This efficiency facilitates the application of our algorithm to extensive datasets, where traditional methods become computationally impractical.

3.2 Optimization

In this section we cover in detail how to solve the optimization problem of Eq. (3). We use a four-step alternate optimization algorithm that solves one variable while fixing the others.

• Solving $\mathbf{W}_i$ with fixed $\mathbf{A}$, $\mathbf{G}$ and α_i.

In such a case, the optimization problem of Eq. (3) w.r.t. $\mathbf{W}_i$ can be simplified to

$$\min_{\mathbf{W}_i} \sum_{i=1}^{v} {\alpha_i}^2 \|\mathbf{X}_i - \mathbf{W}_i\mathbf{A}\mathbf{G}\|_{\mathbf{F}}^2, \quad \text{s.t. } \mathbf{W}_i^\top\mathbf{W}_i = \mathbf{I}_k. \tag{4}$$

Now, to eliminate terms unrelated to $\mathbf{W}_i$, we expand the Frobenius norm into trace form, and Eq. (4) is equivalent to

$$\min_{\mathbf{W}_i} \sum_{i=1}^{v} \alpha_i^2 \mathrm{Tr}((\mathbf{X}_i - \mathbf{W}_i\mathbf{A}\mathbf{G})(\mathbf{X}_i - \mathbf{W}_i\mathbf{A}\mathbf{G})^\top). \tag{5}$$

Then the objective function can be converted to

$$\max_{\mathbf{W}_i} \mathrm{Tr}(\mathbf{W}_i^\top \mathbf{C}_i), \quad \text{s.t. } \mathbf{W}_i^\top \mathbf{W}_i = \mathbf{I}_k, \tag{6}$$

where $\mathbf{C}_i = \mathbf{X}_i \mathbf{G}^\top \mathbf{A}^\top$. Finally, we perform Singular Value Decomposition(SVD) for $\mathbf{C}_i$, the optimal $\mathbf{W}_i$ can be obtained by calculating $\mathbf{U}\mathbf{V}^\top$ according to [27], where $\mathbf{U}$ and $\mathbf{V}^\top$ are the left and right singular values of $\mathbf{C}_i$ respectively .

The time complexity required to compute each $\mathbf{C}_i$ is $\mathcal{O}(d_i nk + d_i k^2)$, since there are v views of $\mathbf{C}_i$ to compute, calculating $\mathbf{C}$ needs $\mathcal{O}(dnk + dk^2)$ in all. Additionally, solving Eq. (6) needs $\mathcal{O}(k^2 \sum_{i=1}^{v} d_i) = O(k^2 d)$, so the total time complexity of solving $\mathbf{W}$ is $\mathcal{O}(dnk + dk^2)$.

• Solving $\mathbf{A}$ with fixed $\mathbf{W}_i$, $\mathbf{G}$ and α_i.

The objective function w.r.t. $\mathbf{A}$ arrives at

$$\min_{\mathbf{A}} \sum_{i=1}^{v} {\alpha_i}^2 \|\mathbf{X}_i - \mathbf{W}_i \mathbf{A}\mathbf{G}\|_{\mathbf{F}}^2, \quad \text{s.t. } \mathbf{A}^\top \mathbf{A} = \mathbf{I}_k. \tag{7}$$

Analogous to the approach employed for $\mathbf{W}_i$, the above Eq. (7) is equivalently reformulated into an optimization task prioritizing trace maximization:

$$\max_{\mathbf{A}} \mathrm{Tr}(\mathbf{A}^\top \mathbf{D}), \quad \text{s.t. } \mathbf{A}^\top \mathbf{A} = \mathbf{I}_k, \tag{8}$$

where $\mathbf{D}$ is defined as $\mathbf{D} = \sum_{i=1}^{v} \alpha_i^2 \mathbf{W}_i^\top \mathbf{X}_i \mathbf{G}^\top$. The optimal $\mathbf{A}$ is identified by computing the singular value decomposition of $\mathbf{D}$ and employing its left and right singular vectors. The computational demand for updating $\mathbf{A}$ involves a complexity of $\mathcal{O}(dnk^2 + k^3)$, similar to the strategy for the global anchor matrix $\mathbf{A}$.

• Solving $\mathbf{G}$ with fixed $\mathbf{W}_i$, $\mathbf{A}$ and α_i.

To update the matrix $\mathbf{G}$, while holding matrices $\mathbf{W}_i$, $\mathbf{A}$, and coefficients α_i fixed, we consider an augmented objective that includes a regularization component. The optimization task is formalized as follows:

$$\min_{\mathbf{G}} \sum_{i=1}^{v} \alpha_i^2 \left\|\mathbf{X}_i - \mathbf{W}_i \mathbf{A}\mathbf{G}\right\|_{\mathbf{F}}^2 + \lambda \left\|\mathbf{G}\right\|_{\mathbf{F}}^2, \quad \text{s.t. } \mathbf{G} \geq 0, \mathbf{G}^\top \mathbf{1} = \mathbf{1}. \tag{9}$$

The optimization framework for $\mathbf{G}$ can be rephrased as a Quadratic Programming (QP) formulation:

$$\min_{\mathbf{G}_{:,j}} \frac{1}{2} \mathbf{G}_{:,j}^\top \mathbf{M} \mathbf{G}_{:,j} + \mathbf{f}^\top \mathbf{G}_{:,j}, \quad \text{s.t. } \mathbf{G}_{:,j}^\top \mathbf{1} = 1, \mathbf{G}_{:,j} \geq 0, \tag{10}$$

where the matrix $\mathbf{M}$ and vector $\mathbf{f}$ are intricately defined to incorporate the regularization effect. Specifically, matrix $\mathbf{N}$ is first computed as $\mathbf{N} = 2\sum_{i=1}^{v} \alpha_i \mathbf{I} + 2\lambda \mathbf{I}$, which is then symmetrized to form $\mathbf{M}$ by averaging it with its transpose, $\mathbf{M} = (\mathbf{N} + \mathbf{N}^\top)/2$. Vector $\mathbf{f}$ is determined by aggregating the contributions from all views, $\mathbf{f} = -2\sum_{i=1}^{v} \mathbf{X}_{i[:,j]}^\top \mathbf{W}_i \mathbf{A}$.

This iterative method is realized by executing the QP solver for each row of $\mathbf{G}$, which is a k-dimensional vector, and hence the overall time complexity for this sub-problem is $\mathcal{O}(nk^3)$.

• Solving α_i with fixed $\mathbf{W}_i$, $\mathbf{A}$ and $\mathbf{G}$.

The objective is to minimize the weighted sum of squared norms subject to a normalization constraint on the coefficients. Formally, the problem is stated as follows:

$$\min_{\boldsymbol{\alpha}} \sum_{i=1}^{v} \alpha_i^2 \xi_i^2, \quad \text{s.t. } \boldsymbol{\alpha}^\top \mathbf{1} = 1, \alpha_i \geq 0, \tag{11}$$

where ξ_i represents the Frobenius norm of the residual matrix for the i-th view, expressed as $\xi_i = \|\mathbf{X}_i - \mathbf{W}_i \mathbf{A}\mathbf{G}\|_{\mathbf{F}}$. Employing the Cauchy-Schwarz inequality, an analytic solution for the optimal α_i is derived as:

$$\alpha_i = \frac{\frac{1}{\xi_i}}{\sum_{i=1}^{v} \frac{1}{\xi_i}}. \tag{12}$$

Algorithm 1 AUF-LMC

1: **Input:** Input v views dataset $\{\mathbf{X}_i\}_{i=1}^{v}$, the number of cluster k and λ.
2: **Initialize:** Initialize $\mathbf{W}, \mathbf{A}, \mathbf{G}$. Initialize α_i with $\frac{1}{v}$.
3: **while** not converged **do**
4: Solving $\mathbf{W}_i$ by Eq. (6).
5: Solving $\mathbf{A}$ by Eq. (8).
6: Solving $\mathbf{G}$ by Eq. (10).
7: Solving α_i by Eq. (12).
8: **end while**
9: Get the spectral embedding $\mathbf{H}$ by performing SVD on $\mathbf{G}$.

The whole procedure of AUF-LMC is summarized in Algorithm 1. We use the spectral embedding $\mathbf{H}$ and k-means to obtain the final clustering result.

3.3 Time complexity analysis

The computational complexity of the proposed model is comprehensively analyzed by dissecting each step of the optimization process.

• Complexity of solving $\mathbf{W}_i$.

The time complexity required to compute the matrix $\mathbf{C}_i = \mathbf{X}_i \mathbf{G}^\top \mathbf{A}^\top$ is $\mathcal{O}(d_i nk + d_i k^2)$. The Singular Value Decomposition (SVD) of $\mathbf{C}_i$ requires $\mathcal{O}(d_i k^2)$ time, given that $\mathbf{C}_i$ is a $d_i \times k$ matrix and the dominant cost is for k singular values and vectors. The subsequent multiplication of matrices to update $\mathbf{W}_i$ necessitates $\mathcal{O}(d_i k^2)$ operations, considering the product of a $d_i \times k$ and a $k \times k$ matrix.

• Complexity of solving $\mathbf{A}$.

The time complexity required to compute the matrix $\mathbf{D} = \sum_{i=1}^{v} \alpha_i^2 \mathbf{W}_i^\top \mathbf{X}_i \mathbf{Z}^\top$ is $\mathcal{O}(d_i nk + nk^2)$. The SVD needed for updating $\mathbf{A}$ has a computational complexity of $\mathcal{O}(k^3)$, due to the decomposition of a $k \times k$ matrix $\mathbf{D}$. Additional $\mathcal{O}(k^3)$ time is required for the matrix multiplication involved in this step.

• Complexity of solving $\mathbf{G}$.

Solving the Quadratic Programming (QP) problem for $\mathbf{G}$ has a complexity of $\mathcal{O}(nk^3)$, which accrues for each column across all samples, dominated by the inversion required in the optimization process.

• Complexity of solving α_i.

The computation of α_i is a direct operation and hence incurs only $\mathcal{O}(1)$ complexity.

Aggregating the complexities from each step, the total time cost T for the optimization phase can be expressed as:

$$T = \sum_{i=1}^{v} \mathcal{O}(d_i nk + d_i k^2) + \mathcal{O}(d_i nk + nk^2 + k^3) + \mathcal{O}(nk^3) + \mathcal{O}(1).$$

This summation simplifies to $\mathcal{O}((d + d_i)nk + dk^2 + nk^2 + k^3 + nk^3)$, where $d = \sum_{i=1}^{v} d_i$ is the dimension of all views. Assuming that k are significantly smaller than n and d_i, the complexity can be approximated as $\mathcal{O}(nk^3)$, showing a quasi-linear relationship with the number of samples, thus reinforcing the overall efficiency of the algorithm.

4 Experiments

4.1 Experimental Setup

Benchmark Datasets. In the experimental evaluation of multi-view clustering algorithms, a diverse range of datasets are pivotal for a comprehensive analysis. The datasets employed in our study are shown in Table 1. MNIST is a large-scale handwritten numeric data set that is widely used in many fields[17],[21],[29].

Table 1. The multi-view datasets used in the experiment.

Dataset	Samples	View	Class	Features
Caltech101-20[14]	2,386	6	20	48/40/254/1,984/512/928
Caltech101-all[6]	9,144	5	102	48/40/254/512/928
CCV	6,773	3	20	20/20/20
NUSWIDEOBJ[5]	30,000	5	31	65/226/145/74/129
MNIST	60,000	3	10	342/1,024/64

Table 2. The clustering results of various multi-view clustering algorithms were compared, the evaluation indexes include ACC, NMI, Purity and F-score, the higher the better. Bold in the table indicates the best result, and '-' indicates out of memory.

Dataset	Caltech101-20									
Metric \ Method	PMSC	mPAC	MLRSSC	AMGL	SFMC	RMKM	BMVC	LMVSC	FPMVS	Ours
ACC	0.5981	0.4983	0.3600	0.1876	0.5947	0.3961	0.1769	0.4304	0.6547	**0.6706**
NMI	0.5244	0.5855	0.2008	0.1101	0.5641	0.5034	0.1708	0.5553	0.6326	**0.6450**
Purity	0.7125	0.6622	0.4476	0.6313	0.7045	0.3150	0.4166	0.7125	0.7368	**0.7519**
Fscore	0.4329	0.4806	0.3069	0.4661	0.4303	0.2799	0.1197	0.3414	0.6905	**0.6990**
	Caltech101-all									
ACC	-	0.2031	0.1365	0.0359	0.1777	0.1650	0.2123	0.2005	0.3015	**0.3199**
NMI	-	0.3809	0.1066	0.0187	0.2613	0.3494	**0.4246**	0.4155	0.3549	0.4162
Purity	-	0.2914	0.1371	0.4311	0.2430	0.0875	**0.4124**	0.3975	0.3460	0.3788
Fscore	-	0.1254	0.0815	**0.3617**	0.0462	0.0548	0.1854	0.1586	0.2326	0.3057
	CCV									
ACC	-	0.2311	0.1259	0.1102	0.1156	0.1584	0.1326	0.2014	0.2399	**0.2410**
NMI	-	0.1744	0.0471	0.0758	0.0346	0.1136	0.0763	0.1657	0.1760	**0.1808**
Purity	-	**0.2917**	0.1307	0.2021	0.1194	0.1902	0.1652	0.2396	0.2605	0.2653
Fscore	-	0.1346	0.1095	0.1215	0.1085	0.1084	0.0826	0.1194	**0.1419**	0.1417
	NUSWIDEOBJ									
ACC	-	-	-	-	0.1221	0.1193	0.1299	0.1583	0.1946	**0.1961**
NMI	-	-	-	-	0.0095	0.0926	0.1290	0.1337	0.1351	**0.1354**
Purity	-	-	-	-	0.1227	0.2062	0.2333	**0.2488**	0.2382	0.2376
Fscore	-	-	-	-	0.1140	0.0750	0.0881	0.0990	0.1372	**0.1372**
	MNIST									
ACC	-	-	-	-	-	0.8621	0.4595	0.9852	0.9884	**0.9884**
NMI	-	-	-	-	-	0.9209	0.3959	0.9576	0.9651	**0.9651**
Purity	-	-	-	-	-	0.8988	0.4766	0.9852	0.9884	**0.9884**
Fscore	-	-	-	-	-	0.8728	0.3357	0.9704	0.9768	**0.9768**

Compared Multi-View Clustering Algorithms We benchmark our algorithm against a suite of nine established multi-view clustering techniques: PMSC[11], mPAC[10], MLRSSC[1], AMGL[22], SFMC[13], RMKM[3], BMVC[31], LMVSC[12] and FPMVS[28]. In these methods, PMSC, mPAC, and MLRSSC are categorized under classic subspace-based multi-view clustering techniques. LMVSC, FPMVS, along with our proposed method, are categorized under anchor-based subspace methodologies. Lastly, methods like SFMC, RMKM, and BMVC represent approaches that incorporate bipartite graphs, k-means, or binarization to address the challenges of multi-view clustering on a large scale.

4.2 Experimental Result

Clustering Performance The comprehensive evaluation of our algorithm demonstrates exceptional performance in the realm of multi-view clustering, as detailed in Table 2. Our approach significantly surpasses competing methodologies across the various datasets examined. This is particularly evident in the Caltech101-20 dataset, where our algorithm achieved the highest scores in all metrics, reinforcing its capability to discern and adapt to the intrinsic cluster structures within the data.Furthermore, when considering the larger-scale

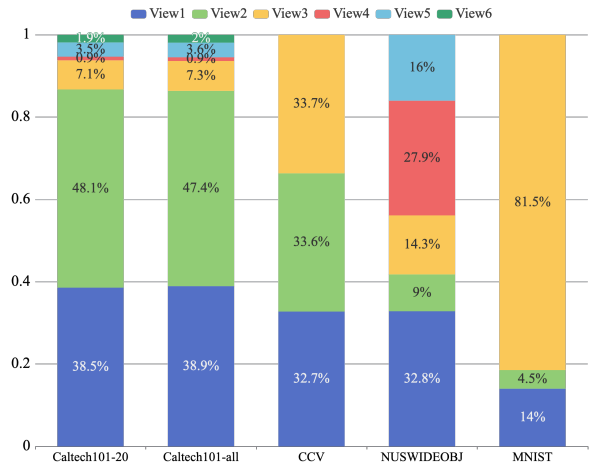

Fig. 2. Visualization of learned view coefficients on individual datasets.

datasets such as CCV and NUS-WIDE-OBJ, our method sustains high performance, indicative of its robustness and adaptability to different data complexities. It is worth noting that the MNIST dataset, which presents its unique challenges due to high dimensionality, saw our approach achieving impressive results that underscore its effectiveness in feature distinction critical for clustering tasks.

The consistency of these results across multiple datasets attests to the algorithm's versatility and confirms its potential for practical applications in various multi-view learning scenarios. The improvements in clustering performance, particularly noted in ACC, demonstrate the algorithm's capacity to leverage latent information across multiple views efficiently.

The Learned View Coefficients Fig. 2 illustrates the learned view coefficients for each of the five benchmark datasets. The distribution of weights across different views highlights the contributions of each view to the overall clustering performance. In datasets such as Caltech101-20 and Caltech101-all, the weight distribution is relatively even among the first three views, suggesting that these views contribute similarly to capturing the data's intrinsic structure. In contrast, for the MNIST dataset, view 3 seems to dominate, indicating that this view might carry more discriminative features that are crucial for clustering in this particular dataset.Overall, the visualization of learned view weights underscores our algorithm's ability to adaptively assign importance to different views based on their relevance and contribution to the clustering task. This adaptive learning of view-specific weights is pivotal, as it not only enhances clustering accuracy but also offers insights into the significance of each view within the context of multi-view learning. The results across various datasets confirm the versatility of our approach, demonstrating its capability to handle diverse and complex multi-view data efficiently.

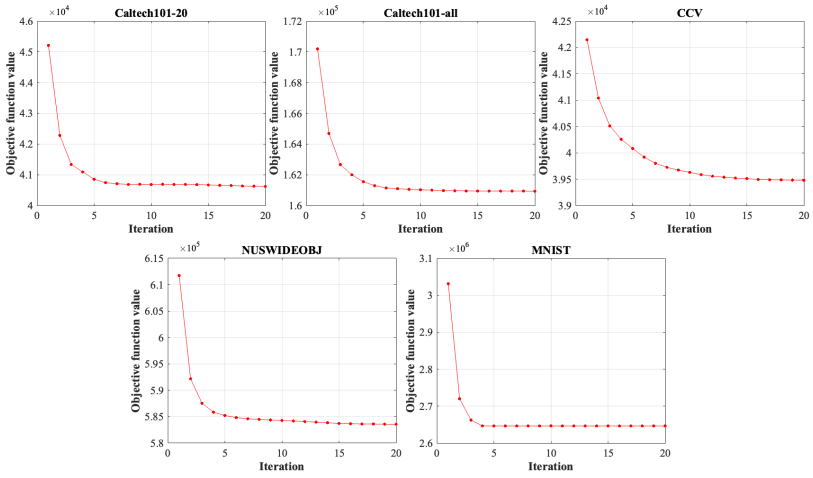

Fig. 3. Visualization of convergence of the objective function on various datasets.

Convergence The convergence performance of AUF-LMC is visualized across five benchmark datasets, as shown in the appended Fig. 3. From the observed trends, it is evident that our method achieves rapid convergence, which signifies the efficiency of the learning-based anchor selection and the robustness of the optimization process.Across the datasets, the objective function values exhibit a sharp decline in the initial iterations, followed by a plateau, indicating that our method swiftly approaches an optimal or near-optimal solution. This quick stabilization is a testament to the algorithm's ability to effectively integrate the complementary information from different views and the efficacy of the regularization term in achieving a consistent clustering structure.

5 Conclusion

In this paper, a novel multi-view clustering method termed Adaptive Unified Framework with Global Anchor Graph for Large-scale Multi-view Clustering(AUF-LMC) based on global anchor graph is proposed. AUF-LMC integrates learning-based anchor selection with multi-view bipartites to enhance multi-view clustering and better mine the underlying structure of data. We design a unified framework, which greatly improves the consistency and complementarity between different views, and enhances the generalization ability of our model through a series of constraint strategies. In addition, we design a detailed optimization method for the algorithm and prove its high efficiency in theory. Our extensive experimental validation on multiple standard datasets shows that our approach not only stands out in terms of clustering accuracy, but also demonstrates significant efficiency and scalability. These results confirm that AUF-LMC is well suited to handle complex large-scale multi-view data analysis and pave the way for future practical applications in various fields.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of Jiangsu Province under Grant BK20221379, in part by the National Natural Science Foundation of China under Grant 62306048, in part by the Changzhou University CNPC-CZU Innovation Alliance under Grant CCIA2023-01; in part by the Changzhou Leading Innovative Talent Introduction and Cultivation Project under Grant 20221460; in part by Changzhou Applied Basic Research Fund Project Grant (CQ20230092; CJ20235036).

References

1. Brbić, M., Kopriva, I.: Multi-view low-rank sparse subspace clustering. Pattern Recognition **73**, 247–258 (2018)
2. Cai, D., Chen, X.: Large scale spectral clustering via landmark-based sparse representation. IEEE transactions on cybernetics **45**(8), 1669–1680 (2014)
3. Cai, X., Nie, F., Huang, H.: Multi-view k-means clustering on big data. In: Twenty-Third International Joint conference on artificial intelligence (2013)
4. Chen, D., Li, J., Guizilini, V., Ambrus, R.A., Gaidon, A.: Viewpoint equivariance for multi-view 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9213–9222 (2023)
5. Chua, T.S., Tang, J., Hong, R., Li, H., Luo, Z., Zheng, Y.: Nus-wide: a real-world web image database from national university of singapore. In: Proceedings of the ACM international conference on image and video retrieval. pp. 1–9 (2009)
6. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In: 2004 conference on computer vision and pattern recognition workshop. pp. 178–178. IEEE (2004)
7. Gui, Z., Yang, J., Xie, Z.: Learning an enhanced consensus representation for multi-view clustering via latent representation correlation preserving. Knowledge-Based Systems **253**, 109479 (2022)
8. Huang, D., Wang, C.D., Lai, J.H.: Fast multi-view clustering via ensembles: Towards scalability, superiority, and simplicity. IEEE Transactions on Knowledge and Data Engineering (2023)
9. Jiang, G., Peng, J., Wang, H., Mi, Z., Fu, X.: Tensorial multi-view clustering via low-rank constrained high-order graph learning. IEEE Transactions on Circuits and Systems for Video Technology **32**(8), 5307–5318 (2022)
10. Kang, Z., Guo, Z., Huang, S., Wang, S., Chen, W., Su, Y., Xu, Z.: Multiple partitions aligned clustering. arXiv preprint arXiv:1909.06008 (2019)
11. Kang, Z., Zhao, X., Peng, C., Zhu, H., Zhou, J.T., Peng, X., Chen, W., Xu, Z.: Partition level multiview subspace clustering. Neural Networks **122**, 279–288 (2020)
12. Kang, Z., Zhou, W., Zhao, Z., Shao, J., Han, M., Xu, Z.: Large-scale multi-view subspace clustering in linear time. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 4412–4419 (2020)
13. Li, X., Zhang, H., Wang, R., Nie, F.: Multiview clustering: A scalable and parameter-free bipartite graph fusion method. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(1), 330–344 (2020)

14. Li, Y., Nie, F., Huang, H., Huang, J.: Large-scale multi-view spectral clustering via bipartite graph. In: Proceedings of the AAAI conference on artificial intelligence. vol. 29 (2015)
15. Liang, W., Zhou, S., Xiong, J., Liu, X., Wang, S., Zhu, E., Cai, Z., Xu, X.: Multi-view spectral clustering with high-order optimal neighborhood laplacian matrix. IEEE Transactions on Knowledge and Data Engineering **34**(7), 3418–3430 (2020)
16. Liu, M., Yang, Z., Li, L., Li, Z., Xie, S.: Auto-weighted collective matrix factorization with graph dual regularization for multi-view clustering. Knowledge-Based Systems **260**, 110145 (2023)
17. Liu, Y., Cheng, D., Zhang, D., Xu, S., Han, J.: Capsule networks with residual pose routing. IEEE Transactions on Neural Networks and Learning Systems pp. 1–14 (2024). https://doi.org/10.1109/TNNLS.2023.3347722
18. Liu, Y., Dong, X., Zhang, D., Xu, S.: Deep unsupervised part-whole relational visual saliency. Neurocomputing p. 126916 (2023)
19. Liu, Y., Zhang, D., Liu, N., Xu, S., Han, J.: Disentangled capsule routing for fast part-object relational saliency. IEEE Transactions on Image Processing **31**, 6719–6732 (2022)
20. Liu, Y., Zhang, D., Zhang, Q., Han, J.: Part-object relational visual saliency. IEEE transactions on pattern analysis and machine intelligence **44**(7), 3688–3704 (2021)
21. Liu, Y., Zhou, L., Wu, G., Xu, S., Han, J.: Tcgnet: Type-correlation guidance for salient object detection. IEEE Transactions on Intelligent Transportation Systems (2023)
22. Nie, F., Li, J., Li, X., et al.: Parameter-free auto-weighted multiple graph learning: a framework for multiview clustering and semi-supervised classification. In: IJCAI. vol. 9, pp. 1881–1887 (2016)
23. Sun, M., Wang, S., Zhang, P., Liu, X., Guo, X., Zhou, S., Zhu, E.: Projective multiple kernel subspace clustering. IEEE Transactions on Multimedia **24**, 2567–2579 (2021)
24. Sun, M., Zhang, P., Wang, S., Zhou, S., Tu, W., Liu, X., Zhu, E., Wang, C.: Scalable multi-view subspace clustering with unified anchors. In: Proceedings of the 29th ACM International Conference on Multimedia. pp. 3528–3536 (2021)
25. Wang, M., Fu, W., Hao, S., Tao, D., Wu, X.: Scalable semi-supervised learning by efficient anchor graph regularization. IEEE Transactions on Knowledge and Data Engineering **28**(7), 1864–1877 (2016)
26. Wang, S., Liu, X., Liu, L., Tu, W., Zhu, X., Liu, J., Zhou, S., Zhu, E.: Highly-efficient incomplete large-scale multi-view clustering with consensus bipartite graph. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9776–9785 (2022)
27. Wang, S., Liu, X., Zhu, E., Tang, C., Liu, J., Hu, J., Xia, J., Yin, J.: Multi-view clustering via late fusion alignment maximization. In: IJCAI. pp. 3778–3784 (2019)
28. Wang, S., Liu, X., Zhu, X., Zhang, P., Zhang, Y., Gao, F., Zhu, E.: Fast parameter-free multi-view subspace clustering with consensus anchor guidance. IEEE Transactions on Image Processing **31**, 556–568 (2021)
29. Xu, S., Gu, J., Hua, Y., Liu, Y.: Dktnet: Dual-key transformer network for small object detection. Neurocomputing **525**, 29–41 (2023)
30. Yan, W., Xu, J., Liu, J., Yue, G., Tang, C.: Bipartite graph-based discriminative feature learning for multi-view clustering. In: Proceedings of the 30th ACM International Conference on Multimedia. pp. 3403–3411 (2022)
31. Zhang, Z., Liu, L., Shen, F., Shen, H.T., Shao, L.: Binary multi-view clustering. IEEE transactions on pattern analysis and machine intelligence **41**(7), 1774–1782 (2018)