

Capturing Individuality and Commonality Between Anchor Graphs for Multi-View Clustering

Zhoumin Lu^{1,2}, Yongbo Yu¹, Linru Ma^{2*}, Feiping Nie^{1*}, Rong Wang¹

¹School of Computer Science, School of Artificial Intelligence, OPtics and ElectroNics (iOPEN), Northwestern Polytechnical University, Xi'an 710072, China.

²Institute of Systems Engineering, AMS, Beijing 100071, China.

walker.zhoumin.lu@gmail.com, yuyongbo122@qq.com,
malinru@163.com, feipingnie@gmail.com, wangrong07@tsinghua.org.cn

Abstract

The use of anchors often leads to better efficiency and scalability, making them highly favored. However, there is a challenge in anchor-based multi-view subspace learning. A unified anchor graph overly emphasize the commonality between views, failing to adequately capture the view-specific individuality. This has led some models to independently explore the individuality of each view before aligning and integrating them, often achieving better performance but making the process more cumbersome. Therefore, this paper proposes a new model, simultaneously capturing the individuality and commonality between anchor graphs for multi-view clustering. The model has three notable advantages: First, it allows view-specific anchor graphs to align in real-time with a common anchor graph as a reference, eliminating the need for post-alignment. Second, it enforces a cluster-wise structure among anchors and balances sample distribution among them, providing strong discriminative power. Lastly, it maintains linear complexity with respect to the numbers of samples and anchors, avoiding the significant time costs associated with their increase. Comprehensive experiments demonstrate the effectiveness and efficiency of our method compared to various state-of-the-art algorithms.

1 Introduction

Clustering has long been a crucial aspect of data analysis, which groups data according to specific patterns without label information. The emergence of multi-source data [Baltrušaitis *et al.*, 2018] has raised higher demands for clustering tasks, giving rise to multi-view extension [Li *et al.*, 2018; Fang *et al.*, 2023]. In this setting, data often comes with multiple perspectives or representations. The consistent and complementary information among these views is explored to address the limitations of single-view information and enhance performance.

*Corresponding author

Multi-view clustering can be roughly categorized into subspace-based [Gao *et al.*, 2020; Huang *et al.*, 2022], kernel-based [Li *et al.*, 2022a; Liu, 2022], matrix factorization-based [Wan *et al.*, 2023], and deep learning-based [Xu *et al.*, 2022; Yan *et al.*, 2023] methods. Subspace clustering is widely recognized as an effective data analysis technique, as it can directly learn affinities between samples, revealing the internal structure of the data. Unfortunately, it often struggles with the high computational costs associated with large-scale datasets.

On the other hand, the use of anchors has brought about better efficiency and scalability, making them highly popular. Naturally, this technology has been integrated into subspace learning models. In anchor-based subspace learning, a point-to-anchor graph is learned instead of a point-to-point graph. Since the number of anchors is usually much smaller than that of samples, it is more friendly to both runtime and memory. Anchors can be pre-selected, resorting to heuristic strategies such as k-means and feature scoring. However, these pre-selection strategies face the risk of being insufficient to represent the entire dataset, which leads to dynamic anchor learning strategies.

However, a challenge arises in anchor-based multi-view subspace learning. A unified anchor graph tends to overemphasize the commonality between views, failing to adequately capture the individuality of each view. This has led some models to independently explore each view's individuality before aligning and integrating them, often achieving better performance but making the process more cumbersome. Moreover, although existing models maintain linear complexity with respect to the number of samples, they often exhibit quadratic or even cubic complexity with respect to the number of anchors. The growth of anchors inevitably results in substantial time costs.

To address these issues, this paper proposes a new model, Capturing Individuality and Commonality between Anchor Graphs (CICAG) for multi-view clustering. The four contributions are summarized as follows:

- It captures both the individuality and commonality between anchor graphs, addressing the limitations of previous approaches that focus solely on either aspect.
- It enables view-specific anchor graphs to align in real-time with a common anchor graph as a reference, eliminating the need for post-alignment.

- It enforces a cluster-wise structure among anchors and balances samples distribution among them, providing strong discriminative power.
- It achieves linear computational complexity concerning both the number of samples and anchors, making it efficient and scalable even with a large number of anchors.

2 Related Work

Anchor-based multi-view learning models can be categorized into two main types: static anchor pre-selection and dynamic anchor learning. The former determine anchors in advance using heuristic methods, with no changes during subsequent anchor graph construction. The latter requires joint optimization of anchors and anchor graphs with dynamic updates. Dynamic anchor learning includes unified anchor graph strategies, which share a common anchor graph across all views, and view-specific anchor graph strategies, which learn a separate anchor graph for each view.

LMVSC [Kang *et al.*, 2020] employs k-means to pre-select anchors for each view and constructs non-unified anchor graphs, which are then concatenated across views. SFMC [Li *et al.*, 2022b] and TBGL [Xia *et al.*, 2022] pre-select aligned anchors by concatenating all features and using a feature scoring strategy, subsequently merging multiple anchor graphs. Unlike SFMC, TBGL utilizes tensors to further explore complementary information across views.

FPMVS-CAG [Wang *et al.*, 2022b], OMSC [Chen *et al.*, 2022], EMVGC-LG [Wen *et al.*, 2023], and CAMVC [Zhang *et al.*, 2024] are representative models, dynamically exploring a unified anchor graph. FPMVS-CAG projects all views into a latent shared space to learn a single anchor graph. OMSC builds on FPMVS-CAG by jointly learning the mapping from the anchor graph to the cluster indicator matrix. EMVGC-LG embeds both local and global structural information into the anchor graph. CAMVC uses pseudo-labels to enforce cluster structure distribution among anchors.

FMVACC [Wang *et al.*, 2022a], FDAGF [Zhang *et al.*, 2023], and AEVC [Liu *et al.*, 2024] are based on anchor-based subspace learning models, capturing separate anchor graphs for different views. FMVACC and AEVC both employ post-alignment strategies, whereas FDAGF directly concatenates anchor graphs from each view. Unlike FMVACC, which averages after alignment, AEVC embeds the learned anchors and anchor graphs into a new model for secondary learning, serving as regularization.

3 Methodology

3.1 Formulation

Typically, the framework for single-view subspace learning takes the following form.

$$\min_{\mathbf{A}, \mathbf{Z}} \|\mathbf{X} - \mathbf{A}\mathbf{Z}\|_F^2 + \gamma \|\mathbf{Z}\|_F^2 \quad (1)$$

where $\mathbf{X} \in \mathbb{R}^{d \times n}$ represents the complete single-view dataset, $\mathbf{A} \in \mathbb{R}^{d \times m}$ is a representative subset of samples, $\mathbf{Z} \in \mathbb{R}^{m \times n}$ is the weight matrix used to linearly combine \mathbf{A} to reconstruct \mathbf{X} , n , m and d are the number of samples,

anchors and features, respectively. More concisely, \mathbf{A} and \mathbf{Z} can be referred to as the anchors and anchor graph, respectively. The anchor graph describes an affinity between \mathbf{X} and \mathbf{A} . As a multi-view extension strategy, a unified anchor graph is often required, resulting in the following form.

$$\min_{\mathbf{A}^{(i)}, \mathbf{Z}} \sum_{i=1}^v \left\| \mathbf{X}^{(i)} - \mathbf{A}^{(i)} \mathbf{Z} \right\|_F^2 + \gamma \|\mathbf{Z}\|_F^2 \quad (2)$$

where \mathbf{Z} is the anchor graph shared across views, while $\mathbf{X}^{(i)}$ and $\mathbf{A}^{(i)}$ denote the samples and anchors of the i -th view, respectively. Regrettably, this form overly emphasizes the commonality between views and fails to adequately capture the individuality of each view, thus giving rise to another form.

$$\min_{\mathbf{A}^{(i)}, \mathbf{Z}^{(i)}} \sum_{i=1}^v \left\| \mathbf{X}^{(i)} - \mathbf{A}^{(i)} \mathbf{Z}^{(i)} \right\|_F^2 + \gamma \|\mathbf{Z}^{(i)}\|_F^2 \quad (3)$$

where $\mathbf{Z}^{(i)}$ is the anchor graph unique to the i -th view. This case independently explores the individuality of each view before aligning and integrating them, often achieving better performance, but also making the process more cumbersome. Therefore, we propose a compromise solution.

$$\begin{aligned} \min_{\mathbf{A}^{(i)}, \mathbf{Z}^{(i)}, \mathbf{Z}} & \sum_{i=1}^v \left\| \mathbf{X}^{(i)} - \mathbf{A}^{(i)} \left[\alpha \mathbf{Z}^{(i)} + (1 - \alpha) \mathbf{Z} \right] \right\|_F^2 \\ & + \gamma \left(\|\mathbf{Z}^{(i)}\|_F^2 + \|\mathbf{Z}\|_F^2 \right) \end{aligned} \quad (4)$$

In this approach, we not only extract a common \mathbf{Z} for all views but also exploit a specific $\mathbf{Z}^{(i)}$ for each view. The anchor set $\mathbf{A}^{(i)}$ are shared between $\mathbf{Z}^{(i)}$ and \mathbf{Z} , ensuring that $\mathbf{Z}^{(i)}$ and \mathbf{Z} are well-aligned. Indirectly, through the assistance of \mathbf{Z} , this alignment relationship is shared among $\mathbf{Z}^{(i)}$.

Meanwhile, the anchors are desired to be evenly distributed across the clusters and further aligned. Thus, the following strategy is adopted.

$$\max_{\mathbf{Z}^{(i)}, \mathbf{Z}} \sum_{i=1}^v \text{Tr} \left(\mathbf{F}^T \mathbf{Z}^{(i)} \mathbf{Z}^T \mathbf{F} \right) \quad (5)$$

where $\mathbf{F} \in \mathbb{R}^{m \times c}$ is a predefined anchor-to-cluster assignment matrix, with each cluster containing at least $\lfloor m/c \rfloor$ and at most $\lceil m/c \rceil$ anchors. For instance, if there are 4 anchors and 2 clusters, then \mathbf{F} is set as

$$\mathbf{F} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \quad (6)$$

Next, we further consider the constraints regarding several variables. If $\mathbf{A}^{(i)}$ can effectively represent $\mathbf{X}^{(i)}$, the following equality holds.

$$\mathbf{X}^{(i)} = \mathbf{A}^{(i)} \left(\mathbf{A}^{(i)T} \mathbf{A}^{(i)} \right)^{-1} \mathbf{A}^{(i)T} \mathbf{X}^{(i)} \quad (7)$$

This equation implies a series of simpler non-trivial solutions, which necessitate the orthogonality of $\mathbf{A}^{(i)}$.

$$\Omega_{\mathbf{A}^{(i)}} = \left\{ \mathbf{A}^{(i)} | \mathbf{A}^{(i)T} \mathbf{A}^{(i)} = \mathbf{I} \right\} \quad (8)$$

It is suggested to more freely retain the individuality of $\mathbf{Z}^{(i)}$, only with each row summing to 1.

$$\Omega_{\mathbf{Z}^{(i)}} = \left\{ \mathbf{Z}^{(i)} | \mathbf{Z}^{(i)} \geq \mathbf{0}, \mathbf{Z}^{(i)T} \mathbf{1} = 1 \right\} \quad (9)$$

Meanwhile, \mathbf{Z} must ensure a more balanced distribution of samples among anchors, subject to the following constraint.

$$\Omega_{\mathbf{Z}} = \left\{ \mathbf{Z} | \mathbf{Z} \geq \mathbf{0}, \mathbf{Z}^T \mathbf{1} = \mathbf{1}, \mathbf{Z} \mathbf{1} = \mathbf{l}, \mathbf{l} = \frac{n}{m} \mathbf{1} \right\} \quad (10)$$

Considering Problems 4-5 and Constraints 8-10, the final objective function is formulated as follows:

$$\begin{aligned} & \min_{\mathbf{A}^{(i)}, \mathbf{Z}^{(i)}, \mathbf{Z}} \sum_{i=1}^v \left\| \mathbf{X}^{(i)} - \mathbf{A}^{(i)} \left[\alpha \mathbf{Z}^{(i)} + (1-\alpha) \mathbf{Z} \right] \right\|_F^2 \\ & - \beta \operatorname{Tr} (\mathbf{F}^T \mathbf{Z}^{(i)} \mathbf{Z}^T \mathbf{F}) + \gamma \left(\left\| \mathbf{Z}^{(i)} \right\|_F^2 + \left\| \mathbf{Z} \right\|_F^2 \right) \\ & \text{s.t. } \mathbf{A}^{(i)} \in \Omega_{\mathbf{A}^{(i)}}, \mathbf{Z}^{(i)} \in \Omega_{\mathbf{Z}^{(i)}}, \mathbf{Z} \in \Omega_{\mathbf{Z}} \end{aligned} \quad (11)$$

Figure 1 illustrates our proposed CICAG framework. It dynamically captures an individual anchor graph $\mathbf{Z}^{(i)}$ for each view's data $\mathbf{X}^{(i)}$ while simultaneously exploring a common anchor graph \mathbf{Z} for all views. $\mathbf{Z}^{(i)}$ and \mathbf{Z} share a dynamic anchor set $\mathbf{A}^{(i)}$, thus enabling indirect real-time alignment between individual anchor graphs. Meanwhile, a predefined anchor-to-cluster assignment matrix \mathbf{F} enforces the cluster-wise structure among anchors and enhances alignment. Additionally, the common anchor graph requires a more balanced distribution of samples across the anchors.

3.2 Optimization

To solve our model, the following alternating optimization strategy is employed.

Update $\mathbf{A}^{(i)}$

If the components unrelated to $\mathbf{A}^{(i)}$ are removed and $\mathbf{W}^{(i)}$ is precomputed as

$$\mathbf{W}^{(i)} = \mathbf{X}^{(i)} \left[\alpha \mathbf{Z}^{(i)T} + (1-\alpha) \mathbf{Z}^T \right] \quad (12)$$

the resulting $\mathbf{A}^{(i)}$ -subproblem is given.

$$\max_{\mathbf{A}^{(i)}} \operatorname{Tr} (\mathbf{A}^{(i)T} \mathbf{W}^{(i)}) \quad \text{s.t. } \mathbf{A}^{(i)T} \mathbf{A}^{(i)} = \mathbf{I} \quad (13)$$

Furthermore, the closed-form solution for Equation (13) can be revealed by Theorem 1 [Liu *et al.*, 2021].

Theorem 1. Suppose that $\mathbf{W}^{(i)} = \mathbf{U}_c^{(i)} \boldsymbol{\Sigma}_c^{(i)} \mathbf{V}_c^{(i)T}$, where $\mathbf{U}_c^{(i)} \in \mathbb{R}^{d_i \times c}$, $\boldsymbol{\Sigma}_c^{(i)} \in \mathbb{R}^{c \times c}$ and $\mathbf{V}_c^{(i)} \in \mathbb{R}^{m \times c}$. Problem 13 has a closed-form solution $\mathbf{A}^{(i)} = \mathbf{U}_c^{(i)} \mathbf{V}_c^{(i)T}$.

Update $\mathbf{Z}^{(i)}$

If the components unrelated to $\mathbf{Z}^{(i)}$ are removed, and $\tilde{\mathbf{E}}^{(v)}$ and $\mathbf{E}^{(v)}$ are precomputed as

$$\tilde{\mathbf{E}}^{(i)} = (1-\alpha) \mathbf{Z} - \mathbf{A}^{(i)T} \mathbf{X}^{(i)} \quad (14)$$

$$\mathbf{E}^{(i)} = \left(2\alpha \tilde{\mathbf{E}}^{(i)} - \beta \mathbf{F} \mathbf{F}^T \mathbf{Z} \right) / (\alpha^2 + \gamma) \quad (15)$$

the resulting $\mathbf{Z}^{(i)}$ -subproblem is given.

$$\begin{aligned} & \min_{\mathbf{Z}^{(i)}} \operatorname{Tr} (\mathbf{Z}^{(i)T} \mathbf{Z}^{(i)}) + \operatorname{Tr} (\mathbf{Z}^{(i)T} \mathbf{E}^{(i)}) \\ & \text{s.t. } \mathbf{Z}^{(i)} \geq \mathbf{0}, \mathbf{Z}^{(i)T} \mathbf{1} = 1 \end{aligned} \quad (16)$$

Furthermore, the optimal solution for Equation (16) can be revealed by Theorem 2.

Theorem 2. By introducing the Lagrange multiplier $\mu^{(i)}$ and predefining $\mathbf{S}^{(i)}$ and $\Phi(\mathbf{S}^{(i)})$ as

$$\mathbf{S}^{(i)} = \mathbf{1} \mu^{(i)T} - \mathbf{E}^{(i)} \quad (17)$$

$$\Phi(\mathbf{S}^{(i)}) = \max_{\mathbf{Z}^{(i)} \geq \mathbf{0}} \operatorname{Tr} (\mathbf{Z}^{(i)T} \mathbf{S}^{(i)}) - \operatorname{Tr} (\mathbf{Z}^{(i)T} \mathbf{Z}^{(i)}) \quad (18)$$

the dual problem of Problem 16 is derived as

$$\min_{\mu^{(i)}} \Phi(\mathbf{S}^{(i)}) - \mu^{(i)T} \mathbf{1} \quad (19)$$

whose exact gradient with respect to $\mu^{(i)}$ is given by

$$\frac{\partial L}{\partial \mu^{(i)}} = \frac{1}{2} [\mathbf{S}^{(i)}]_+^T \mathbf{1} - \mathbf{1} \quad (20)$$

Once the optimal $\mu^{(i)}$ is found, the desired $\mathbf{Z}^{(i)}$ can be recovered by

$$\mathbf{Z}^{(i)} = [\mathbf{1} \mu^{(i)T} - \mathbf{E}^{(i)}]_+ / 2 \quad (21)$$

Proof. Similar to the proof of Theorem 3 but simpler. \square

Update \mathbf{Z}

If the components unrelated to \mathbf{Z} are removed, and $\tilde{\mathbf{M}}^{(i)}$ and \mathbf{M} are precomputed as

$$\tilde{\mathbf{M}}^{(i)} = \alpha \mathbf{Z}^{(i)} - \mathbf{A}^{(i)T} \mathbf{X}^{(i)} \quad (22)$$

$$\mathbf{M} = \frac{\sum_{i=1}^v \left[2(1-\alpha) \tilde{\mathbf{M}}^{(i)} - \beta \mathbf{F} \mathbf{F}^T \mathbf{Z}^{(i)} \right]}{v [(1-\alpha)^2 + \gamma]} \quad (23)$$

the resulting \mathbf{Z} -subproblem is given.

$$\begin{aligned} & \min_{\mathbf{Z}} \operatorname{Tr} (\mathbf{Z}^T \mathbf{Z}) + \operatorname{Tr} (\mathbf{Z}^T \mathbf{M}) \\ & \text{s.t. } \mathbf{Z} \geq \mathbf{0}, \mathbf{Z}^T \mathbf{1} = \mathbf{1}, \mathbf{Z} \mathbf{1} = \mathbf{l} \end{aligned} \quad (24)$$

Furthermore, the optimal solution for Equation (24) can be revealed by Theorem 3.

Theorem 3. By introducing the Lagrange multipliers μ and ν , and predefining \mathbf{S} and $\Phi(\mathbf{S})$ as

$$\mathbf{S} = (\mathbf{1} \mu^T + \nu \mathbf{1}^T - \mathbf{M}) \quad (25)$$

$$\Phi(\mathbf{S}) = \max_{\mathbf{Z} \geq \mathbf{0}} \operatorname{Tr} (\mathbf{Z}^T \mathbf{S}) - \operatorname{Tr} (\mathbf{Z}^T \mathbf{Z}) \quad (26)$$

the dual problem of Problem 24 is

$$\min_{\mu, \nu} \Phi(\mathbf{S}) - \mu^T \mathbf{1} - \nu^T \mathbf{l} \quad (27)$$

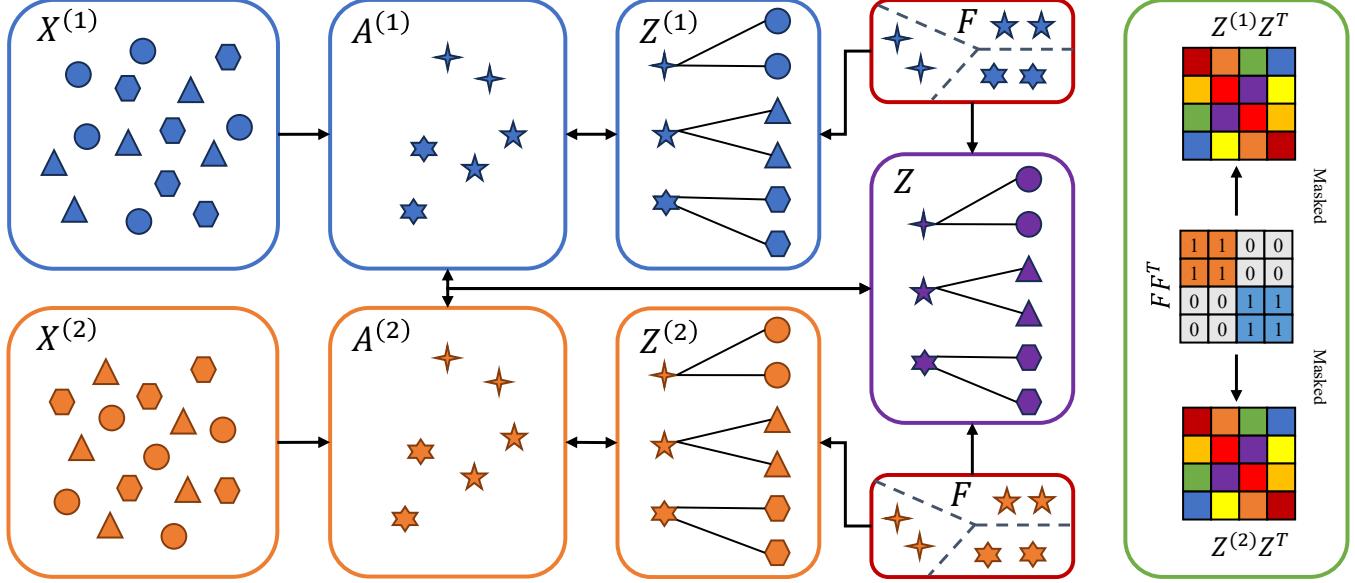


Figure 1: Illustration of our CICAG framework.

whose exact gradients with respect to μ and ν are given by

$$\frac{\partial L}{\partial \mu} = \frac{1}{2} [S^{(i)}]_+^T \mathbf{1} - \mathbf{1} \quad (28)$$

$$\frac{\partial L}{\partial \nu} = \frac{1}{2} [S^{(i)}]_+ \mathbf{1} - l \quad (29)$$

Once the optimal μ and ν are found, the desired Z can be recovered by

$$Z = [\mathbf{1}\mu^T + \nu\mathbf{1}^T - M]_+/2 \quad (30)$$

Proof. The following min-max problem is given

$$\min_{z_j \geq 0} \max_{\mu, \nu} \sum_{j=1}^n z_j^T m_j + z_j^T z_j - \mu^T (Z^T \mathbf{1} - \mathbf{1}) - \nu^T (Z \mathbf{1} - l) \quad (31)$$

which is easily transformed into the max-min form.

$$\max_{\mu, \nu} \mu^T \mathbf{1} + \nu^T l + \sum_{j=1}^n \min_{z_j \geq 0} z_j^T (m_j - \nu - \mu_j \mathbf{1}) + z_j^T z_j \quad (32)$$

Further, there are the following max-max problems

$$\max_{\mu, \nu} \mu^T \mathbf{1} + \nu^T l - \sum_{j=1}^n \max_{z_j \geq 0} z_j^T (\nu + \mu_j \mathbf{1} - m_j) - z_j^T z_j \quad (33)$$

Obviously, Equation (33) is equivalent to Equation (27). In addition, split $\Phi(S)$ by column as

$$\phi(s_j) = \max_{z_j \geq 0} z_j^T s_j - z_j^T z_j \quad (34)$$

it is easy to konw the derivatives as

$$\begin{aligned} \nabla \phi(s_j) &= \arg \max_{z_j \geq 0} z_j^T s_j - z_j^T z_j \\ &= \arg \max_{z_j \geq 0} \|z_j - \frac{s_j}{2}\|_2^2 = \frac{[s_j]_+}{2} \end{aligned} \quad (35)$$

Further, its gradients w.r.t. μ and ν can be computed. \square

Final Fusion

Once our model converges, the desired anchor graph is obtained by Equation (36). For convenience, our methodological procedure is summarized in Algorithm 1.

$$Z^* = \frac{\alpha}{v} \sum_{i=1}^v Z^{(i)} + (1 - \alpha) Z \quad (36)$$

Algorithm 1 CICAG Solver

Input: Dataset $\{X^{(i)}\}_{i=1}^v$, anchor number m , cluster number c , and parameters α, β and γ .

Output: Learned anchor graph Z^* .

- 1: Initialize $Z^{(i)}$, Z and F .
 - 2: **while** non-convergence **do**
 - 3: Update $A^{(i)}$ by Theorem 1.
 - 4: Update $Z^{(i)}$ by Theorem 2.
 - 5: Update Z by Theorem 3.
 - 6: **end while**
 - 7: Obtain Z^* by Equation (36).
-

3.3 Complexity

Time Complexity. In the $A^{(i)}$ -subproblem, computing $W^{(i)}$ and obtaining its singular value decomposition necessitates $\mathcal{O}(nmd + mdc)$, while updating $A^{(i)}$ only costs $\mathcal{O}(mdc)$. In the $Z^{(i)}$ -subproblem, computing $\tilde{E}^{(i)}$ and $E^{(i)}$ together involves $\mathcal{O}(nmd + nmc)$, obtaining the gradient with respect to $\mu^{(i)}$ entails $\mathcal{O}(nm)$, and updating $\mu^{(i)}$ and $Z^{(i)}$ involves $\mathcal{O}(n)$ and $\mathcal{O}(nm)$ respectively. In the Z -subproblem, computing M demands $\mathcal{O}(nmd + nmcv)$, obtaining the gradients with respect to μ and ν both necessitate $\mathcal{O}(nm)$, and updating μ , ν , and Z entails $\mathcal{O}(n)$, $\mathcal{O}(m)$, and $\mathcal{O}(nm)$ respec-

tively. Thus, in one iteration, our model requires a computational complexity of only $\mathcal{O}(nmd + nmcv + mdc)$, which implies that our algorithm is not only linear in the number of samples n , but also linear in the number of anchors m , data dimension d , and the number of clusters c . Table 1 lists some notable algorithms whose complexities often reach quadratic or even cubic with respect to m , which actually limits their scalability. Additionally, the complexity of some algorithms with respect to c is not linear, making them less suitable for data with a large number of clusters.

Method	Computational Complexity
LMVSC	$\mathcal{O}(nmc + nmd + mdc)$
SFMC	$\mathcal{O}(nm^2 + nmd + nmc + m^3)$
TBGL	$\mathcal{O}(nm^2 + nm \log n + nmc + nmd)$
FPMVS	$\mathcal{O}(nm^3 + nmd)$
OMSC	$\mathcal{O}(nm^3 + m^2c + mc^3 + nmc + dc^3)$
FMVACC	$\mathcal{O}(nm^2 + nmd + m^2d + m^3)$
FDAGF	$\mathcal{O}(nm^2 + nmd + m^2d)$
EMVGCLG	$\mathcal{O}(nm^3 + nmd)$
AEVC	$\mathcal{O}(nm^3 + nmd)$
CAMVC	$\mathcal{O}(nm^2c^2 + nmdc + m^3c^3 + m^2c^2d)$
CICAG	$\mathcal{O}(nmd + nmc + mdc)$

Table 1: The computational complexity of state-of-the-art various advanced algorithms.

Space Complexity. The original data $\{\mathbf{X}^{(i)}\}$ consumes $\mathcal{O}(nd)$ space. The auxiliary variables generated from updating $\mathbf{A}^{(i)}$ necessitate $\mathcal{O}(md)$, and $\mathbf{A}^{(i)}$ also takes up $\mathcal{O}(md)$. The auxiliary variables generated from updating $\mathbf{Z}^{(i)}$ require $\mathcal{O}(nmv)$, and $\mathbf{Z}^{(i)}$ also takes up $\mathcal{O}(nmv)$. The auxiliary variables generated from updating \mathbf{Z} demand $\mathcal{O}(nm)$ space, and \mathbf{Z} itself also uses $\mathcal{O}(nm)$. Therefore, our algorithm needs only $\mathcal{O}(nd + md + nmv)$ space complexity, which is typical for anchor-based methods.

3.4 Convergence

Although the overall objective function, involving multiple variables, is non-convex, fortunately, each subproblem concerning a single variable is convex. For the $\mathbf{A}^{(i)}$ -subproblem, a closed-form solution exists. For the $\mathbf{Z}^{(i)}$ -subproblem and the \mathbf{Z} -subproblem, their equivalent dual problems can be constructed. Since the exact gradients of the dual problems can be obtained, any gradient-based method can be used to find their optimal solutions. Let the objective function value at the t -th iteration be $\mathcal{J}(\{\mathbf{A}_t^{(i)}\}, \{\mathbf{Z}_t^{(i)}\}, \mathbf{Z}_t)$, then the following inequality holds.

$$\begin{aligned} \mathcal{J}(\{\mathbf{A}_t^{(i)}\}, \{\mathbf{Z}_t^{(i)}\}, \mathbf{Z}_t) &\geq \mathcal{J}(\{\mathbf{A}_{t+1}^{(i)}\}, \{\mathbf{Z}_t^{(i)}\}, \mathbf{Z}_t) \\ &\geq \mathcal{J}(\{\mathbf{A}_{t+1}^{(i)}\}, \{\mathbf{Z}_{t+1}^{(i)}\}, \mathbf{Z}_t) \geq \mathcal{J}(\{\mathbf{A}_{t+1}^{(i)}\}, \{\mathbf{Z}_{t+1}^{(i)}\}, \mathbf{Z}_{t+1}) \\ &\geq \dots \geq (2\gamma/m - \beta) nv \end{aligned} \quad (37)$$

Obviously, since the function value monotonically decreases with each iteration and has a lower bound, this algorithm can ensure convergence to a local minimum.

4 Experiment

4.1 Experimental Protocols

Datasets. Our experiments employ 9 public datasets for comparison, including 3Sources, WebKB, NUS-WIDE, Notting-Hill, Cifar10, Cifar100, YouTubeFace10, YouTubeFace20 and YouTubeFace50. Specifically, the last five are large-scale datasets. Their statistics are described in Table 2.

Dataset	#Views	#Instances	#Classes	#Features
3Sources	3	169	3	3560, 3631, 3068
WebKB	2	1051	2	2949, 334
NUS-WIDE	5	3000	25	64, 225, 144, 73, 128
Notting-Hill	3	4660	5	6750, 3304, 2000
Cifar10	3	50000	10	512, 2048, 1024
Cifar100	3	50000	100	512, 2048, 1024
YouTubeFace10	4	38654	10	944, 576, 512, 640
YouTubeFace20	4	63896	20	944, 576, 512, 640
YouTubeFace50	4	126054	50	944, 576, 512, 640

Table 2: The statistics for multi-view datasets.

Baselines. Our CICAG is compared with the following 9 algorithms: SFMC [Li *et al.*, 2022b], TBGL [Xia *et al.*, 2022], FPMVS [Wang *et al.*, 2022b], OMSC [Chen *et al.*, 2022], FMVACC [Wang *et al.*, 2022a], FDAGF [Zhang *et al.*, 2023], EMVG [Wen *et al.*, 2023], CAMVC [Zhang *et al.*, 2024] and AEVC [Liu *et al.*, 2024]. These classic or state-of-the-art algorithms are highly correlated with ours and have a near-linear computational complexity.

Settings. For our model, α is set to 0.1, while the remaining hyperparameters are tuned by a grid search, whose ranges are $m \in \{1c, 3c, 5c\}$, $\beta \in \{0.001, 0.01, \dots, 100, 1000\}$, and $\gamma \in \{0.001, 0.01, \dots, 100, 1000\}$. The hyperparameters of all compared algorithms are set by grid search according to the recommendations of their original papers. Considering the initialization sensitivity and randomness during running, we performed all algorithms 10 times.

Metrics. In our experiments, the performance is evaluated by Accuracy (ACC), Normalized Mutual Information (NMI) and Adjusted Rand Index (ARI). Their means and standard deviations are reported in experiments. Besides, their average runtime is also recorded.

4.2 Experimental Results

Performance. Table 3 shows, CICAG achieves the highest ACC across all used datasets, with its NMI and ARI being either the highest or second highest. Other algorithms show mixed results across different datasets. Additionally, the notably low standard deviation suggests strong algorithmic stability. Overall, our algorithm CICAG is highly competitive.

Runtime. As observed from Table 1, CICAG exhibits linear complexity in terms of the number of samples, anchors, and features. When the number of anchors and features is large, CICAG is more efficient compared to other algorithms. However, since the publicly available datasets have generally undergone preprocessing such as feature extraction or dimensionality reduction, making m and d relatively small, CICAG's efficiency advantage is not as apparent. Nevertheless, Figure 2 shows that, CICAG's efficiency consistently ranks in

Metric	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
Dataset	3Sources			WebKB			NUS-WIDE		
SFMC	71.6±0.0	59.9±0.0	50.0±0.0	78.2±0.0	28.0±0.0	49.0±0.0	10.7±0.0	08.2±0.0	01.2±0.0
TBGL	72.8±0.0	54.8±0.0	45.0±0.0	84.1±0.0	18.7±0.0	29.2±0.0	07.8±0.0	05.2±0.0	08.2±0.0
FPMVS	32.5±0.0	08.3±0.0	04.0±0.0	86.9±0.0	44.6±0.0	53.0±0.0	19.2±0.0	16.4±0.0	06.3±0.0
OMSC	37.9±0.0	14.2±0.0	08.1±0.0	93.0±0.0	52.6±0.0	70.4±0.0	18.1±0.0	16.1±0.0	05.9±0.0
FMVACC	71.0±0.0	62.9±0.0	58.3±0.0	93.2±0.0	54.3±0.0	71.7±0.0	17.8±0.0	15.1±0.0	05.9±0.0
FDAGF	72.8±0.0	51.8±0.0	51.1±0.0	93.3±0.0	53.7±0.0	71.0±0.0	16.9±0.0	16.3±0.0	04.7±0.0
EMVGC	61.0±0.0	35.6±0.0	33.0±0.0	93.9±0.1	57.8±0.0	73.2±0.1	18.2±0.3	17.1±0.3	05.6±0.1
CAMVC	73.4±2.4	66.8±0.6	66.2±1.9	87.5±0.0	30.9±0.0	44.9±0.0	19.1±0.3	19.7±0.3	06.1±0.1
AEVC	70.4±0.0	64.9±0.0	56.0±0.0	81.5±0.0	31.1±0.0	38.7±0.0	18.9±0.0	18.3±0.0	06.2±0.0
CICAG	75.2±3.8	65.5±3.1	59.9±5.7	95.2±0.0	64.8±0.0	78.6±0.0	20.5±0.4	19.2±0.2	06.3±0.2
Dataset	Notting-Hill			Cifar10			Cifar100		
SFMC	88.9±0.0	84.3±0.0	84.7±0.0	98.9±0.0	96.9±0.0	97.5±0.0	OM	OM	OM
TBGL	59.9±0.0	54.5±0.0	36.9±0.0	OM	OM	OM	OM	OM	OM
FPMVS	90.0±0.0	80.0±0.0	79.4±0.0	99.0±0.0	97.3±0.0	97.8±0.0	OM	OM	OM
OMSC	73.2±0.0	67.7±0.0	62.4±0.0	98.0±0.0	95.3±0.0	95.7±0.0	OM	OM	OM
FMVACC	78.6±0.0	69.8±0.0	57.0±0.0	99.2±0.0	97.9±0.0	98.3±0.0	91.7±0.0	98.0±0.0	92.1±0.0
FDAGF	84.5±0.0	79.8±0.0	79.2±0.0	98.7±0.0	96.6±0.0	97.2±0.0	85.4±0.0	96.7±0.0	86.6±0.0
EMVGC	84.0±0.1	79.4±0.2	78.7±0.2	99.1±0.0	97.5±0.0	98.0±0.0	94.3±0.8	98.8±0.2	95.1±3.8
CAMVC	95.2±0.0	96.2±0.0	90.5±0.0	99.5±0.0	98.5±0.0	98.9±0.0	91.4±2.1	98.2±0.5	92.8±2.2
AEVC	87.8±0.0	75.0±0.0	75.5±0.0	96.2±0.0	91.5±0.0	91.9±0.0	91.6±0.0	98.1±0.0	92.8±0.0
CICAG	97.6±0.0	94.4±0.1	95.3±0.0	99.5±0.0	98.7±0.0	99.0±0.0	99.9±0.7	99.9±0.1	99.9±0.7
Dataset	YouTubeFace10			YouTubeFace20			YouTubeFace50		
SFMC	63.4±0.0	64.3±0.0	36.3±0.0	58.0±0.0	62.5±0.0	23.4±0.0	OM	OM	OM
TBGL	OM								
FPMVS	75.0±0.0	78.7±0.0	67.1±0.0	OM	OM	OM	OM	OM	OM
OMSC	77.7±0.0	81.2±0.0	67.8±0.0	OM	OM	OM	OM	OM	OM
FMVACC	76.6±0.0	77.1±0.0	69.4±0.0	72.5±0.0	77.7±0.0	52.5±0.0	72.8±0.0	82.8±0.0	62.8±0.0
FDAGF	74.5±0.0	76.3±0.0	63.2±0.0	69.8±0.0	77.4±0.0	57.2±0.0	67.9±0.0	80.6±0.0	56.6±0.0
EMVGC	70.6±0.3	79.2±1.2	67.6±0.6	69.0±1.7	76.2±0.3	59.0±0.1	69.1±2.2	80.2±0.5	61.7±1.1
CAMVC	74.6±0.0	76.2±0.0	65.0±0.0	69.9±1.4	77.6±0.3	60.8±1.0	70.2±1.9	80.8±0.5	59.0±2.4
AEVC	75.6±0.0	78.2±0.0	68.6±0.0	71.9±0.0	75.9±0.0	61.0±0.0	72.3±0.0	82.8±0.0	63.5±0.0
CICAG	81.7±3.1	80.7±1.8	71.4±2.9	73.4±0.8	77.9±0.4	61.8±2.6	76.7±0.2	83.8±0.3	62.3±2.0

Table 3: Accuracy (ACC%±Std%), Normalized Mutual Information (NMI%±Std%) and Adjusted Rand Index (ARI%±Std%) for multi-view clustering algorithm comparison. Best results are in bold. OM indicates that the method suffers out-of-memory error.

	D1	D2	D3	D4	D5	D6	D7	D8	D9
Z	50.9	91.6	18.2	72.4	99.3	94.2	73.3	69.3	72.6
$Z_{\min}^{(i)}$	41.4	51.4	12.9	58.9	84.9	91.3	69.6	65.5	66.7
$Z_{\max}^{(i)}$	57.4	72.5	14.4	87.4	89.2	94.2	75.6	68.9	69.9
$Z_{uf}^{(i)}$	50.3	55.3	14.7	62.9	96.1	94.4	72.2	66.2	72.2
$Z_{af}^{(i)}$	71.0	93.2	17.8	78.6	99.2	91.7	76.6	72.5	72.8
Ours	75.2	95.2	20.5	97.6	99.5	99.9	81.7	73.4	76.7

Table 4: Accuracy (ACC%) for ablation results. D1-D9 mark the datasets used in order. Z and $\{Z^{(i)}\}$ are learned from Equations (2) and (3), $Z_{\min}^{(i)}$ and $Z_{\max}^{(i)}$ are the $Z^{(i)}$ corresponding to the lowest and highest ACC, $Z_{uf}^{(i)}$ and $Z_{af}^{(i)}$ correspond to the unaligned fusion and aligned fusion from $\{Z^{(i)}\}$, respectively.

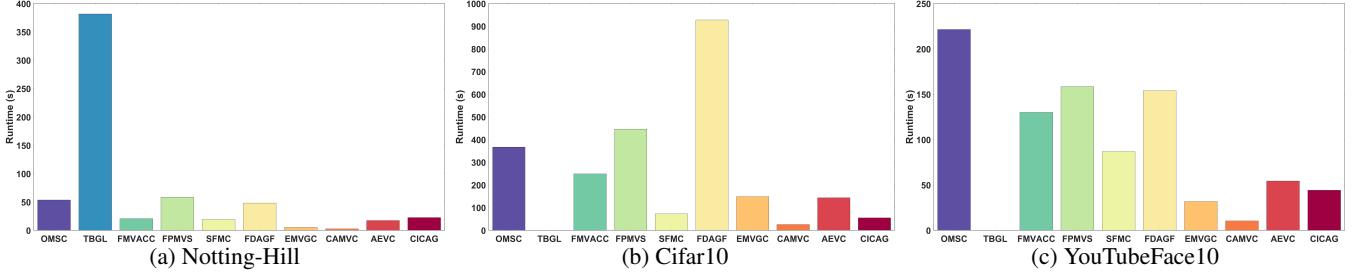


Figure 2: The running time of various algorithms.

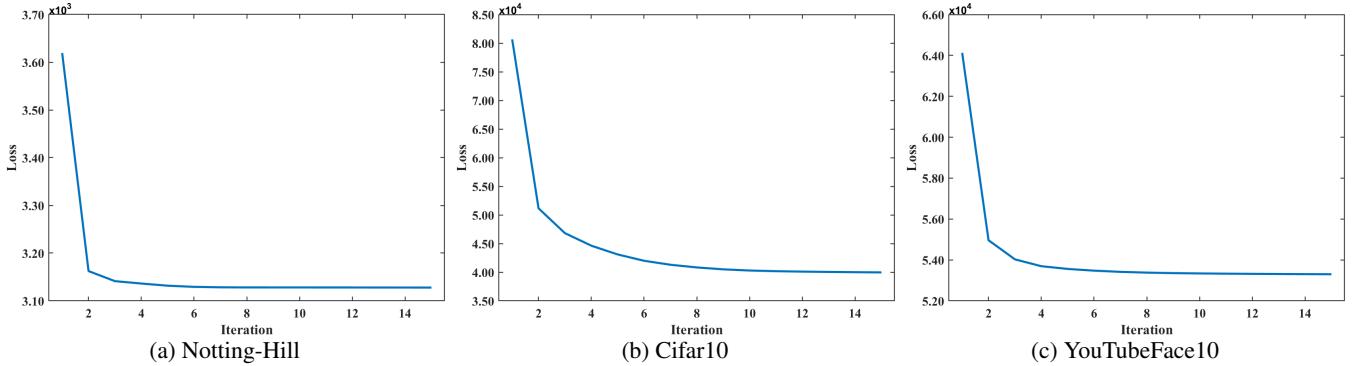
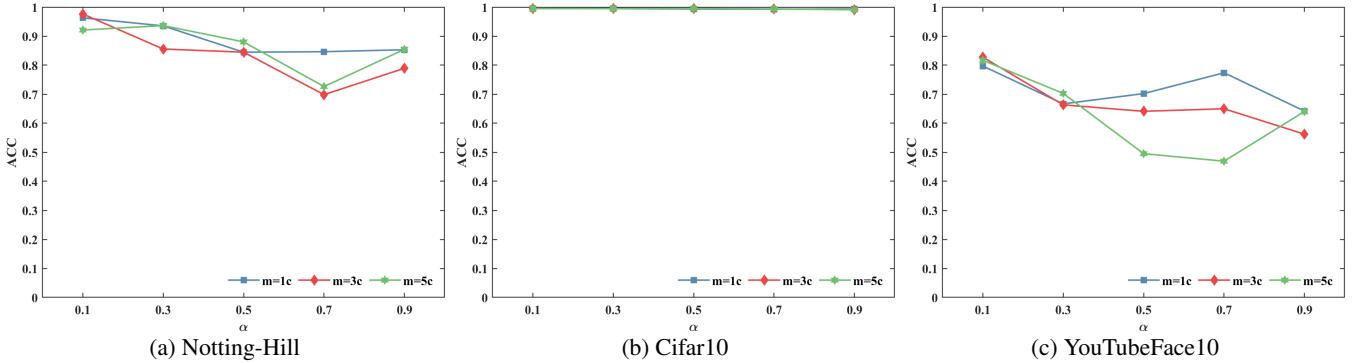
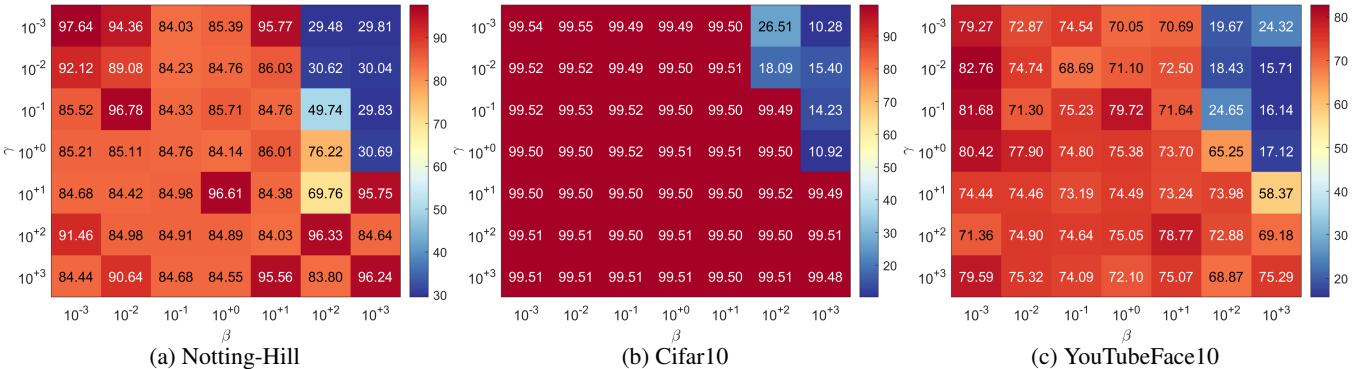


Figure 3: The convergence curves of CICAG.


 Figure 4: The CICAG ACC corresponding to different m and α .

 Figure 5: The CICAG ACC corresponding to different β and γ .

the top three, with only a small difference from the shortest runtime. Considering the performance shown in Table 3, CICAG’s efficiency is entirely acceptable. Although CAMVC requires alternating variable optimization, each variable has a closed-form solution, significantly reducing its runtime. However, once m becomes too large, CAMVC’s computational complexity grows cubically, leading to a significant decrease in efficiency.

Convergence. As seen from Figure 3, the training loss of CICAG on all datasets drops rapidly at first and then stabilizes. This suggests that our algorithm not only converges but also requires only a few iterations.

4.3 Ablation Study

In Table 4, taking the D8 (YouTubeFace20) as an example: the ACC of Z , learned via Equation (2), is 69.3; the ACCs of $Z^{(i)}$, learned via Equation (3), range from 65.5 to 68.9; direct fusion yields an ACC of 66.2; alignment followed by fusion achieves 72.5; and CICAG touches 73.4. These results demonstrate that simultaneously capturing both individuality and commonality is effective.

4.4 Parameter Sensitivity

$m \& \alpha$. As shown in Figure 4, the trend of ACC with varying α is similar across different m , with their maximum values being close. Generally, setting $m = 3c$ is a good choice. As α increases, the ACC of CICAG on NUS-WIDE, Cifar10, Cifar100, YouTubeFace20, and YouTubeFace50 does not show significant changes, while it exhibits a downward trend on WebKB, Notting-Hill, and YouTubeFace10, and a decrease followed by an increase on 3Sources. Overall, α achieves its maximum value around 0.1. Although the ACC at $\alpha = 0.1$ is not optimal for WebKB and YouTubeFace, it is still very close to the maximum value. Therefore, setting $\alpha = 0.1$ is generally a good choice.

$\beta \& \gamma$. As shown in Figure 5, β should generally not exceed γ , while γ itself is relatively insensitive. However, in pursuit of optimal values, neither should be set too high.

5 Conclusion

This paper addresses a challenge in anchor-based multi-view subspace learning. Although multiple views share the same anchor graph to capture the commonality across views, this approach clearly overlooks the individuality of specific views. As a result, some models independently explore the individuality of each view, leading to issues of alignment and fusion of anchor graphs across views. Therefore, this paper proposes a new model, capturing both the individuality and commonality of anchor graphs for multi-view clustering. The model not only allows view-specific anchor graphs to be aligned in real-time based on a common anchor graph, but also generates a balanced sample distribution and cluster-level anchors to enhance discriminative power. Moreover, it is computationally efficient in terms of sample, anchor, feature and cluster numbers. Finally, comprehensive experiments demonstrate that our model is highly competitive in terms of both effectiveness and efficiency compared to other state-of-the-art models.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants 62236001, 62276212, and 62176212.

References

- [Baltrušaitis *et al.*, 2018] Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning: A survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):423–443, 2018.
- [Chen *et al.*, 2022] Man-Sheng Chen, Chang-Dong Wang, Dong Huang, Jian-Huang Lai, and Philip S Yu. Efficient orthogonal multi-view subspace clustering. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 127–135, 2022.
- [Fang *et al.*, 2023] Uno Fang, Man Li, Jianxin Li, Longxiang Gao, Tao Jia, and Yanchun Zhang. A comprehensive survey on multi-view clustering. *IEEE Transactions on Knowledge and Data Engineering*, 35(12):12350–12368, 2023.
- [Gao *et al.*, 2020] Quanxue Gao, Wei Xia, Zhizhen Wan, Deyan Xie, and Pu Zhang. Tensor-svd based graph learning for multi-view subspace clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34(04), pages 3930–3937, 2020.
- [Huang *et al.*, 2022] Shudong Huang, Hongjie Wu, Yazhou Ren, Ivor Tsang, Zenglin Xu, Wentao Feng, and Jiancheng Lv. Multi-view subspace clustering on topological manifold. *Advances in Neural Information Processing Systems*, 35:25883–25894, 2022.
- [Kang *et al.*, 2020] Zhao Kang, Wangtao Zhou, Zhitong Zhao, Junming Shao, Meng Han, and Zenglin Xu. Large-scale multi-view subspace clustering in linear time. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34(04), pages 4412–4419, 2020.
- [Li *et al.*, 2018] Yingming Li, Ming Yang, and Zhongfei Zhang. A survey of multi-view representation learning. *IEEE Transactions on Knowledge and Data Engineering*, 31(10):1863–1883, 2018.
- [Li *et al.*, 2022a] Xingfeng Li, Yinghui Sun, Quansen Sun, and Zhenwen Ren. Consensus cluster center guided latent multi-kernel clustering. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(6):2864–2876, 2022.
- [Li *et al.*, 2022b] Xuelong Li, Han Zhang, Rong Wang, and Feiping Nie. Multiview clustering: A scalable and parameter-free bipartite graph fusion method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):330–344, 2022.
- [Liu *et al.*, 2021] Jiyuan Liu, Xinwang Liu, Yuexiang Yang, Li Liu, Siqi Wang, Weixuan Liang, and Jiangyong Shi. One-pass multi-view clustering for large-scale data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12344–12353, 2021.

- [Liu *et al.*, 2024] Suyuan Liu, Ke Liang, Zhibin Dong, Siwei Wang, Xihong Yang, Sihang Zhou, En Zhu, and Xinwang Liu. Learn from view correlation: An anchor enhancement strategy for multi-view clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26151–26161, 2024.
- [Liu, 2022] Xinwang Liu. SimpleMKKM: Simple multiple kernel k-means. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):5174–5186, 2022.
- [Wan *et al.*, 2023] Xinhang Wan, Xinwang Liu, Jiyuan Liu, Siwei Wang, Yi Wen, Weixuan Liang, En Zhu, Zhe Liu, and Lu Zhou. Auto-weighted multi-view clustering for large-scale data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37(8), pages 10078–10086, 2023.
- [Wang *et al.*, 2022a] Siwei Wang, Xinwang Liu, Suyuan Liu, Jiaqi Jin, Wenxuan Tu, Xinzhong Zhu, and En Zhu. Align then fusion: Generalized large-scale multi-view clustering with anchor matching correspondences. *Advances in Neural Information Processing Systems*, 35:5882–5895, 2022.
- [Wang *et al.*, 2022b] Siwei Wang, Xinwang Liu, Xinzhong Zhu, Pei Zhang, Yi Zhang, Feng Gao, and En Zhu. Fast parameter-free multi-view subspace clustering with consensus anchor guidance. *IEEE Transactions on Image Processing*, 31:556–568, 2022.
- [Wen *et al.*, 2023] Yi Wen, Suyuan Liu, Xinhang Wan, Siwei Wang, Ke Liang, Xinwang Liu, Xihong Yang, and Pei Zhang. Efficient multi-view graph clustering with local and global structure preservation. In *Proceedings of the ACM International Conference on Multimedia*, pages 3021–3030, 2023.
- [Xia *et al.*, 2022] Wei Xia, Quanxue Gao, Qianqian Wang, Xinbo Gao, Chris Ding, and Dacheng Tao. Tensorized bipartite graph learning for multi-view clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):5187–5202, 2022.
- [Xu *et al.*, 2022] Jie Xu, Huayi Tang, Yazhou Ren, Liang Peng, Xiaofeng Zhu, and Lifang He. Multi-level feature learning for contrastive multi-view clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16051–16060, 2022.
- [Yan *et al.*, 2023] Weiqing Yan, Yuanyang Zhang, Chenlei Lv, Chang Tang, Guanghui Yue, Liang Liao, and Weisi Lin. GCFAgg: Global and cross-view feature aggregation for multi-view clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19863–19872, 2023.
- [Zhang *et al.*, 2023] Pei Zhang, Siwei Wang, Liang Li, Changwang Zhang, Xinwang Liu, En Zhu, Zhe Liu, Lu Zhou, and Lei Luo. Let the data choose: Flexible and diverse anchor graph fusion for scalable multi-view clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37(9), pages 11262–11269, 2023.
- [Zhang *et al.*, 2024] Chao Zhang, Xiuyi Jia, Zechao Li, Chunlin Chen, and Huaxiong Li. Learning cluster-wise anchors for multi-view clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38(15), pages 16696–16704, 2024.