One-Step Self-Aligned Anchor Learning for Multi-View Clustering

Zijian Chen[1,*] , **Miao Jia**[1,*] , **Xingchen Hu**[1,†] , **Jiyuan Liu**[1] , **Huan Chen**[1] ,
Siwei Wang[2] and **Jincai Huang**[1]
[1]National University of Defense Technology, Changsha, China
[2]Academy of Military Sciences, Beijing, China
{chenzijian, jiamiao23, chenhuan14, wangsiwei13, huangjincai}@nudt.edu.cn,
xhu4@ualberta.ca, liujiyuan13@163.com

Abstract

Multi-view anchor graph clustering has emerged as a high-efficiency paradigm of multi-view learning. However, how to design an effective anchor alignment mechanism within this framework remains an open challenge, which is formally termed the Anchor-Unaligned Problem (AUP). Current research fails to adequately address two pivotal aspects of this challenge: first, the construction of anchor graphs and the alignment of anchors are two independent stages, overlooking their potential synergistic reinforcement; second, selecting anchors from different views as an alignment baseline often renders the clustering performance highly sensitive to the baseline choice. To address these issues, we propose a unified framework termed One-Step Self-Aligned Anchor Learning for Multi-View Clustering (OSAA-MVC). Departing from conventional two-stage strategies, we integrate anchor alignment and anchor graph construction into a joint optimization process, thereby enabling their mutual reinforcement to improve clustering performance. To avoid the baseline selection issue, we introduce a novel mixed anchor strategy, which effectively bypasses the necessity of manual baseline selection while simultaneously capturing both view-specific and cross-view information. This strategy can stabilize clustering performance at a consistently high level. Extensive experiments demonstrate the superior efficiency and effectiveness of our proposed method compared to state-of-the-art competitors. The code is available at https://github.com/Jiamiao2024/OSAA-MVC.

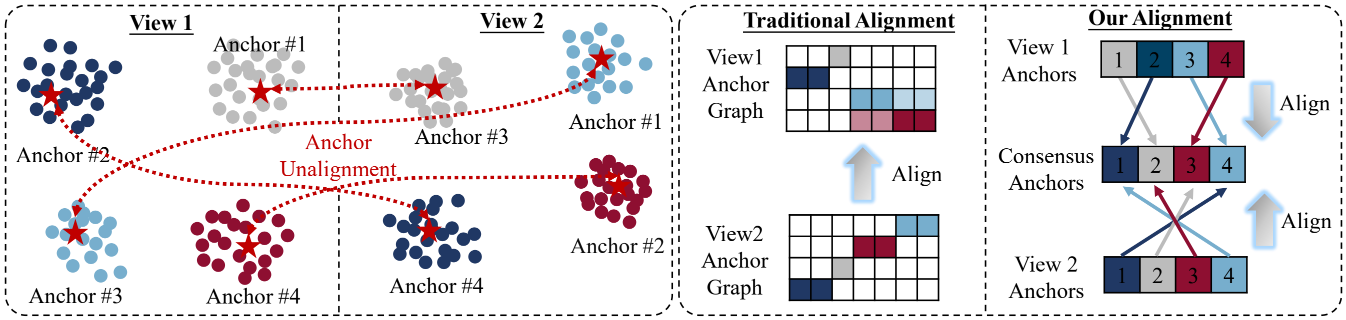
1 Introduction

Real-world data are often collected from multiple sources with different modalities, such as visual signals, textual descriptions, and audio tracks. To discover the underlying knowledge of such data, multi-view clustering (MVC) has emerged as a pivotal unsupervised learning paradigm [Wen *et al.*, 2019]. Among various MVC approaches, multi-view graph clustering (MVGC) is particularly effective [Wen *et al.*, 2022], as it constructs view-specific similarity graphs to capture pairwise relationships and then integrates them into a unified consensus graph.

While MVGC achieves promising performance, its computational complexity is typically $\mathcal{O}(n^3)$ or $\mathcal{O}(n^2)$ with respect to the number of samples, rendering it infeasible for large-scale scenarios. Therefore, multi-view anchor graph clustering (MVAGC) has attracted increasing attention [Kang *et al.*, 2020]. Instead of calculating affinities between all sample pairs, MVAGC selects a small set of representative points, i.e. anchors, to capture the data distribution [Deng *et al.*, 2025]. By constructing an anchor graph between samples and anchors, the complexity is significantly reduced to a linear scale $\mathcal{O}(nm)$, making graph-based clustering feasible for large-scale datasets.

Despite the efficiency of MVAGC, the independent generation of anchors across different views leads to the critical Anchor-Unaligned Problem (AUP) [Wang *et al.*, 2022], as shown in Figure 1 (a). The early studies avoid alignment issues by sampling raw instances with shared indices. For example, SFMC [Li *et al.*, 2020] applies alternate sampling to identify anchors from aggregated similarity scores. However, these methods restrict anchor flexibility. Consequently, recent studies have turned to establishing correct correspondences for flexibly generated anchors across different views. For example, FMVACC [Wang *et al.*, 2022] introduces a post-alignment module that attempts to match anchors based on the similarities of their generated anchor graphs. Building on this, 3AMVC [Ma *et al.*, 2024] further investigates the complex correspondence relationships among varying numbers of anchors across different views.

However, existing multi-view anchor matching approaches suffer from two critical limitations. First, the disjoint optimization of anchor alignment and anchor graph generation inevitably leads to suboptimal clustering performance. In such a disjoint paradigm, independently generated anchors cannot adaptively adjust to subsequent anchor alignment and anchor graph fusion, hindering high-quality consensus graph learning. Second, the indirect alignment strategy based on view-specific anchor graphs is often unstable, as shown in Figure 1(b). This process relies heavily on the quality of the initial anchor graphs. More critically, selecting an optimal baseline view without prior knowledge is extremely challenging. Fixing a single view as the baseline essentially forces other views to conform to its specific distribution. This strategy ignores



(a) Anchor-Unaligned Problem

(b) Difference of Alignment Methods

Figure 1: Illustration of the AUP (Left) and comparison of alignment methods (Right). (a) The anchors generated by two different views may be semantically unaligned. (b) Comparison of alignment strategies: Traditional methods perform indirect alignment by utilizing the anchor graph of a single pivot view as the baseline, making the process highly sensitive to the quality of the baseline graph. In contrast, our OSAA-MVC aligns view-specific anchors with learned consensus anchors, effectively avoiding the dependency on any single view.

the complementarity among different views, which hinders the learning of the consensus anchor graph.

To overcome these limitations, we propose a novel unified framework, named One-Step Self-Aligned Anchor Learning for Multi-View Clustering (OSAA-MVC). The framework of our proposed OSAA-MVC is shown in Figure 2. Instead of existing multi-stage approaches, we integrate anchor generation, anchor alignment, and consensus anchor graph construction into a unified framework. Specifically, we learn view-specific anchors and global consensus anchors in a shared low-dimensional latent space, enabling direct self-alignment during the anchor learning process. This mechanism resolves the dependency on the quality of initial anchor graphs and effectively avoids the suboptimal clustering performance caused by post-alignment. Furthermore, to avoid the dilemma of selecting a robust baseline view, we design a mixed anchor strategy. By fusing the global consensus anchors with aligned view-specific anchors for data reconstruction, our method effectively avoids the bias introduced by a fixed baseline view while capturing both cross-view consistency and view-specific complementarity. Our primary contributions are outlined as follows:

- We propose a unified one-step framework for large-scale multi-view clustering to effectively tackle the Anchor-Unaligned Problem, where anchor alignment and consensus anchor graph generation are seamlessly integrated without separation.
- We design a mixed anchor reconstruction mechanism that fuses the consensus anchors with aligned view-specific anchors. It can effectively avoid the dilemma of baseline view selection.
- We employ a direct anchor self-alignment strategy, which effectively mitigates the dependency on the quality of initial anchor graphs.
- Extensive experiments demonstrate that our algorithm consistently outperforms state-of-the-art competitors.

2 Related Work

2.1 Multi-view Anchor Graph Clustering

To address the scalability bottleneck of traditional graph clustering, multi-view anchor graph clustering (MVAGC) utilizes m representative anchors ($m \ll n$) to construct anchor graphs, reducing the complexity to a linear scale $\mathcal{O}(nm)$ [Ji and Feng, 2023; Xu *et al.*, 2025]. Therefore, the selection of representative anchors is critical.

Traditional methods typically select anchors using static strategies based on random sampling or heuristic algorithms and keep anchors fixed for subsequent graph construction [Shi *et al.*, 2021; Yu *et al.*, 2023]. For example, FMVPG [Shi *et al.*, 2021] adopts direct alternating sampling to obtain anchors. FMCNOF [Yang *et al.*, 2020] uses k-means to obtain cluster centers as anchors, and HBG-MVSC [Zhou *et al.*, 2025] advances the selection process by applying bisecting k-means. The performance of these methods is highly dependent on the quality of the initial anchors.

To address the issue, recent works have adopted dynamic strategies that alternately optimize anchors and anchor graphs to obtain more representative anchors [Sun *et al.*, 2024]. In practice, existing dynamic anchor optimization strategies can be categorized into two main types: consensus anchor learning [Ji and Feng, 2025] and view-specific anchor learning [Liu *et al.*, 2024b; Wen *et al.*, 2023]. The former methods learn a shared anchor set in a latent space and then map it to each view. For example, SMVSC [Sun *et al.*, 2021] learns projected consensus anchors to generate a common anchor graph. MVSC-HFD [Ou *et al.*, 2024] learns a shared anchor set and employs hierarchical dimensional projection matrices to reduce inter-view discrepancies. The consensus anchors may fail to fully capture view-specific information. In contrast, the latter methods learn anchors independently for each view. For example, CAMVC [Zhang *et al.*, 2024] learns anchors separately for each view based on the anchor–cluster assumption. LASD [Zhang *et al.*, 2025] learns anchors with distributions similar to the original data through a multi-view optimal transport framework. However, generating anchors from independent views leads to mismatched anchor sets, which complicates graph fusion.

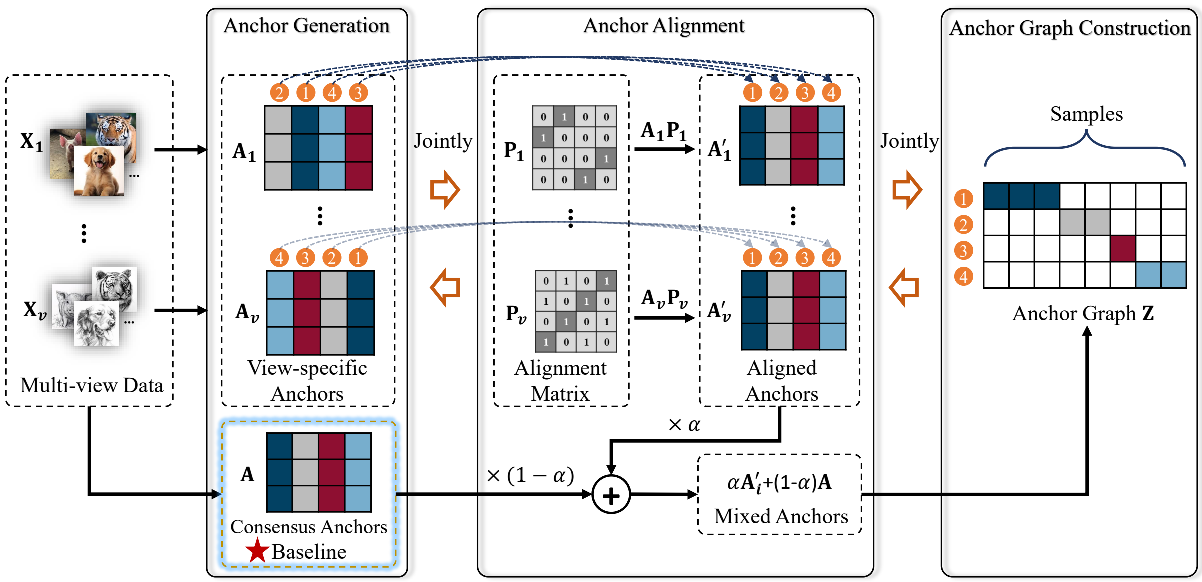


Figure 2: The framework of our proposed OSAA-MVC method. Specifically, in the anchor generation module, OSAA-MVC learns both view-specific anchors and consensus anchors within a shared latent space. In the anchor alignment module, the consensus anchors serve as a baseline to align the view-specific anchors, followed by the generation of mixed anchors. Finally, the consensus anchor graph is constructed utilizing the learned mixed anchors. Notably, these three modules are jointly optimized in a unified framework.

2.2 Anchor-Unaligned Problem

Although anchor strategies are widely adopted to reduce computational complexity, the semantic correspondence of anchors across different views has long been overlooked. A potential solution to AUP is to select raw samples with identical indices across views as anchors [Li *et al.*, 2020]. For example, TBGL [Xia *et al.*, 2022] leverages a variance-based de-correlation anchor selection technique to choose anchors from concatenated view representations. However, this strategy inherently compromises the representational flexibility of the anchors. Recent advances focus on establishing accurate correspondences for flexibly generated anchors. For example, FMVACC [Wang *et al.*, 2022] proposes an anchor alignment module that combines feature and structure information to align anchors across views. However, it simplistically adopts the first view as the alignment baseline, which poses the risk of being influenced by a noisy view. 3AMVC [Ma *et al.*, 2024] further evaluates view quality based on clustering compactness to automatically identify an optimal baseline. This strategy still relies on a single view, neglecting the complementarity among different views. Consequently, AEVC [Liu *et al.*, 2024a] generates a unified anchor graph from concatenated multi-view features to serve as common guidance for alignment, but unfortunately, such simple concatenation overlooks the inherent distribution discrepancies across views.

3 Method

3.1 Formulation

In multi-view learning, a fundamental assumption is that all views share a unified underlying data distribution despite representing different feature spaces [Ren *et al.*, 2021; Dong *et al.*, 2023]. Consequently, it is reasonable to learn a set of representative anchors in a shared latent space. Let $\{\mathbf{X}_i \in \mathbb{R}^{d_i \times n}\}_{i=1}^{v}$ denote a multi-view dataset with n instances and v views, where d_i is the feature dimension of the i-th view. According to [Wang *et al.*, 2021], the general framework for multi-view anchor graph clustering based on global consensus anchor learning can be formulated as

$$\begin{aligned} &\min_{\mathbf{W}_i, \mathbf{A}, \mathbf{Z}} \sum_{i=1}^{v} \|\mathbf{X}_i - \mathbf{W}_i \mathbf{A} \mathbf{Z}\|_F^2 \\ &\text{s.t. } \mathbf{W}_i^\top \mathbf{W}_i = \mathbf{I},\ \mathbf{A}^\top \mathbf{A} = \mathbf{I},\ \mathbf{Z} \geq \mathbf{0},\ \mathbf{Z}^\top \mathbf{1} = \mathbf{1}, \end{aligned} \tag{1}$$

where $\mathbf{W}_i \in \mathbb{R}^{d_i \times k}$ denotes the projection matrix for the i-th view, $\mathbf{A} \in \mathbb{R}^{k \times m}$ represents the consensus anchor matrix in the latent space, $\mathbf{Z} \in \mathbb{R}^{m \times n}$ is the consensus anchor graph, k and m denote the numbers of clusters and anchors, respectively. However, the consensus anchor matrix $\mathbf{A}$ fails to capture view-specific details. We thus introduce view-specific anchors $\mathbf{A}_i \in \mathbb{R}^{k \times m}$ and align them with $\mathbf{A}$ to integrate both common and specific information. Specifically, we obtain the alignment matrix $\mathbf{P}_i$ by minimizing the distance between $\mathbf{A}_i$ and $\mathbf{A}$ as follows

$$\min_{\mathbf{P}_i} \|\mathbf{A}_i \mathbf{P}_i - \mathbf{A}\|_F^2, \quad \text{s.t. } \mathbf{P}_i \in \mathbf{\Omega}_{\mathbf{P}_i}, \tag{2}$$

where $\mathbf{\Omega}_{\mathbf{P}_i}$ denotes the set of permutation matrices defined as

$$\mathbf{\Omega}_{\mathbf{P}_i} = \left\{ \mathbf{P}_i \mid \mathbf{P}_i \mathbf{1} = \mathbf{1},\ \mathbf{P}_i^\top \mathbf{1} = \mathbf{1},\ \mathbf{P}_i \in \{0,1\}^{m \times m} \right\}. \tag{3}$$

By integrating the mixed anchor representation into the reconstruction loss and imposing the alignment constraint, the final objective function is formulated as

$$\begin{aligned}\min_{\mathbf{A}_i,\mathbf{A},\mathbf{Z},\mathbf{W}_i,\mathbf{P}_i} &\sum_{i=1}^{v}\left\|\mathbf{X}_i-\mathbf{W}_i(\alpha\mathbf{A}_i\mathbf{P}_i+(1-\alpha)\mathbf{A})\mathbf{Z}\right\|_F^2\\ &+\sum_{i=1}^{v}\left\|\mathbf{A}_i\mathbf{P}_i-\mathbf{A}\right\|_F^2+\lambda\left\|\mathbf{Z}\right\|_F^2,\\ &\text{s.t. } \mathbf{A}_i^\top\mathbf{A}_i=\mathbf{I},\ \mathbf{A}^\top\mathbf{A}=\mathbf{I},\ \mathbf{W}_i^\top\mathbf{W}_i=\mathbf{I},\\ &\mathbf{Z}^\top\mathbf{1}=\mathbf{1},\ \mathbf{Z}\geq 0,\ \mathbf{P}_i\in\mathbf{\Omega}_{\mathbf{P}_i},\end{aligned} \tag{4}$$

where α is the balance factor for mixed anchors and λ is the regularization hyperparameter. The orthogonality constraints are enforced to enhance the diversity and discriminability of the learned anchors. For the final clustering partition, we directly apply k-means to the right singular vectors of the consensus anchor graph $\mathbf{Z}$ [Guan *et al.*, 2025].

3.2 Optimization

Considering that the objective function in Eq. (4) is not jointly convex with respect to all optimization variables, we adopt an alternating iterative strategy.

Update $\mathbf{A}_i$. Fixing other variables, the optimization problem for $\mathbf{A}_i$ is given by

$$\begin{aligned}\min_{\mathbf{A}_i} &\left\|\mathbf{X}_i-\mathbf{W}_i(\alpha\mathbf{A}_i\mathbf{P}_i+(1-\alpha)\mathbf{A})\mathbf{Z}\right\|_F^2+\left\|\mathbf{A}_i\mathbf{P}_i-\mathbf{A}\right\|_F^2,\\ &\text{s.t. } \mathbf{A}_i^\top\mathbf{A}_i=\mathbf{I}.\end{aligned} \tag{5}$$

The above problem is equivalent to the following trace maximization,

$$\max_{\mathbf{A}_i}\operatorname{Tr}\left(\mathbf{A}_i^\top\mathbf{B}_i\right),\quad \text{s.t. } \mathbf{A}_i^\top\mathbf{A}_i=\mathbf{I}, \tag{6}$$

where $\mathbf{B}_i=\alpha\mathbf{W}_i^\top\mathbf{X}_i\mathbf{Z}^\top\mathbf{P}_i^\top-\alpha(1-\alpha)\mathbf{A}\mathbf{Z}\mathbf{Z}^\top\mathbf{P}_i^\top+\mathbf{A}\mathbf{P}_i^\top$. According to [Hu *et al.*, 2022], this problem admits a closed-form solution, which can be derived using singular value decomposition (SVD).

Update $\mathbf{A}$. Fixing other variables, the sub-problem for the consensus anchor matrix $\mathbf{A}$ can be equivalently transformed into a trace maximization problem as follows

$$\max_{\mathbf{A}}\operatorname{Tr}\left(\mathbf{A}^\top\mathbf{C}\right),\quad \text{s.t. } \mathbf{A}^\top\mathbf{A}=\mathbf{I}, \tag{7}$$

where $\mathbf{C}=\sum_{i=1}^{v}\mathbf{A}_i\mathbf{P}_i-\alpha(1-\alpha)\sum_{i=1}^{v}\mathbf{A}_i\mathbf{P}_i\mathbf{Z}\mathbf{Z}^\top+(1-\alpha)\sum_{i=1}^{v}\mathbf{W}_i^\top\mathbf{X}_i\mathbf{Z}^\top$. Analogous to the optimization of $\mathbf{A}_i$, this problem can be solved by performing SVD on $\mathbf{C}$.

Update $\mathbf{W}_i$. Similarly, the problem can be reformulated in trace form as follows,

$$\max_{\mathbf{W}_i}\operatorname{Tr}\left(\mathbf{W}_i^\top\mathbf{D}_i\right),\quad \text{s.t. } \mathbf{W}_i^\top\mathbf{W}_i=\mathbf{I}, \tag{8}$$

where $\mathbf{D}_i=\mathbf{X}_i\mathbf{Z}^\top(\alpha\mathbf{A}_i\mathbf{P}_i+(1-\alpha)\mathbf{A})^\top$. This problem can be solved by performing SVD on $\mathbf{D}_i$.

Algorithm 1 OSAA-MVC algorithm

Input: Multi-view data matrices $\{\mathbf{X}_i\}_{i=1}^v$, cluster number k, anchor number m, parameters α and λ.
Output: Clustering result.
1: Initialize $\{\mathbf{A}_i\}_{i=1}^v$, $\mathbf{A}$, $\{\mathbf{W}_i\}_{i=1}^v$ as zero matrices, and set $\mathbf{Z}$, $\{\mathbf{P}_i\}_{i=1}^v$ as identity matrices.
2: **while** not converged **do**
3: Update $\{\mathbf{A}_i\}_{i=1}^v$ by solving Eq. (6);
4: Update $\mathbf{A}$ by solving Eq. (7);
5: Update $\{\mathbf{W}_i\}_{i=1}^v$ by solving Eq. (8);
6: Update $\mathbf{Z}$ by solving Eq. (10);
7: Update $\{\mathbf{P}_i\}_{i=1}^v$ by solving Eq. (12);
8: **end while**

Update $\mathbf{Z}$. Fixing other variables, the sub-problem for optimizing $\mathbf{Z}$ is formulated as

$$\begin{aligned}\min_{\mathbf{Z}} &\sum_{i=1}^{v}\left\|\mathbf{X}_i-\mathbf{W}_i(\alpha\mathbf{A}_i\mathbf{P}_i+(1-\alpha)\mathbf{A})\mathbf{Z}\right\|_F^2+\lambda\left\|\mathbf{Z}\right\|_F^2,\\ &\text{s.t. } \mathbf{Z}^\top\mathbf{1}=\mathbf{1},\ \mathbf{Z}\geq 0.\end{aligned} \tag{9}$$

Since the columns of $\mathbf{Z}$ are independent, we can decompose this matrix optimization problem into n separate quadratic programming (QP) tasks. For the j-th column $\mathbf{z}_j$ of $\mathbf{Z}$, the problem is rewritten to

$$\begin{aligned}\min_{\mathbf{z}_j} &\ \frac{1}{2}\mathbf{z}_j^\top\mathbf{G}\mathbf{z}_j+\mathbf{f}^\top\mathbf{z}_j,\\ &\text{s.t. } \mathbf{z}_j^\top\mathbf{1}=1,\ \mathbf{z}_j\geq 0,\end{aligned} \tag{10}$$

where $\mathbf{G}=2(v\alpha^2+v(1-\alpha)^2+\lambda)\mathbf{I}+2\alpha(1-\alpha)(\sum_{i=1}^{v}\mathbf{P}_i^\top\mathbf{A}_i^\top\mathbf{A}+\mathbf{A}^\top\sum_{i=1}^{v}\mathbf{A}_i\mathbf{P}_i)$, $\mathbf{f}^\top=-2\sum_{i=1}^{v}\mathbf{X}_{i[:,j]}^\top\mathbf{W}_i(\alpha\mathbf{A}_i\mathbf{P}_i+(1-\alpha)\mathbf{A})$, and $\mathbf{X}_{i[:,j]}$ represents the j-th column of $\mathbf{X}_i$. The optimal solution is obtained by solving these sub-problems.

Update $\mathbf{P}_i$. Fixing other variables, the optimization problem for updating variable $\mathbf{P}_i$ can be reduced to

$$\begin{aligned}\min_{\mathbf{P}_i} &\left\|\mathbf{X}_i-\mathbf{W}_i\mathbf{A}_{mix}\mathbf{Z}\right\|_F^2+\left\|\mathbf{A}_i\mathbf{P}_i-\mathbf{A}\right\|_F^2,\\ &\text{s.t. } \mathbf{P}_i\mathbf{1}=\mathbf{1},\ \mathbf{P}_i^\top\mathbf{1}=\mathbf{1},\ \mathbf{P}_i\in\{0,1\}^{m\times m},\end{aligned} \tag{11}$$

where $\mathbf{A}_{mix}=\alpha\mathbf{A}_i\mathbf{P}_i+(1-\alpha)\mathbf{A}$. The problem (11) can be equivalently rewritten as

$$\begin{aligned}\max_{\mathbf{P}_i} &\ \operatorname{Tr}(\mathbf{E}_i\mathbf{P}_i),\\ &\text{s.t. } \mathbf{P}_i\mathbf{1}=\mathbf{1},\mathbf{P}_i^\top\mathbf{1}=\mathbf{1},\ \mathbf{P}_i\in\{0,1\}^{m\times m},\end{aligned} \tag{12}$$

where $\mathbf{E}_i=\alpha\mathbf{Z}\mathbf{X}_i^\top\mathbf{W}_i\mathbf{A}_i-\alpha(1-\alpha)\mathbf{Z}\mathbf{Z}^\top\mathbf{A}^\top\mathbf{A}_i+\mathbf{A}^\top\mathbf{A}_i$. According to [Wang *et al.*, 2025], it can be efficiently solved using standard linear programming toolboxes. The overall optimization procedure is summarized in Algorithm 1.

3.3 Discussion

Time Complexity. The optimization process involves updating five variables: $\{\mathbf{A}_i\}_{i=1}^v$, $\mathbf{A}$, $\{\mathbf{W}_i\}_{i=1}^v$, $\mathbf{Z}$, and

Datasets	n	k	v	$d_i(i=1,\ldots,v)$
Yale	165	15	3	4096/3304/6750
Dermatology	358	6	2	12/22
Caltech101-20	2386	20	6	48/40/254/1984/512/928
Scene15	4485	15	3	20/59/40
CCV	6773	20	3	20/20/20
COIL100	7200	100	3	30/99/30
Fashion	10000	10	3	784/784/784
Hdigit	10000	10	2	784/256
VGGFace2	16936	50	4	944/576/512/640
MNIST	60000	10	3	342/1024/64

Table 1: Description of the benchmark multi-view datasets.

$\{\mathbf{P}_i\}_{i=1}^v$. Their corresponding computational complexities are $\mathcal{O}(nmd+vmk^2)$, $\mathcal{O}(nmd+mk^2)$, $\mathcal{O}(nmd+dk^2)$, $\mathcal{O}(nm^3+nmd)$, and $\mathcal{O}(nmd+vm^3)$, respectively, where $d=\sum_{i=1}^v d_i$ represents the total feature dimension. Consequently, the overall complexity is $\mathcal{O}(nmd+nm^3)$. Given that $m \ll n$, the total time cost remains linear with respect to the number of samples n.

Space Complexity. The storage for multi-view data $\{\mathbf{X}_i\}_{i=1}^v$ requires $\mathcal{O}(nd)$. The space complexities for the five optimization variables $\{\mathbf{A}_i\}_{i=1}^v$, $\mathbf{A}$, $\{\mathbf{W}_i\}_{i=1}^v$, $\mathbf{Z}$, and $\{\mathbf{P}_i\}_{i=1}^v$ are $\mathcal{O}(vmk)$, $\mathcal{O}(mk)$, $\mathcal{O}(dk)$, $\mathcal{O}(nm)$, and $\mathcal{O}(vm^2)$, respectively. Consequently, the overall space complexity is $\mathcal{O}(n(d+m)+dk+vm^2)$. The memory requirement is linear with respect to the sample size n, ensuring the proposed OSAA-MVC method is memory-efficient for large-scale applications.

Convergence Analysis. As detailed in the optimization section, we solve each sub-problem optimally by fixing the remaining variables. This iterative updating scheme ensures a monotonic decrease in the objective function value defined in Eq. (4). Therefore, consistent with the findings in [Bezdek and Hathaway, 2003], our proposed algorithm is theoretically guaranteed to converge to a local optimum.

4 Experiments

4.1 Experimental Setting

Datasets. To validate the proposed framework, we conduct extensive experiments on ten widely used multi-view datasets, including Yale, Dermatology, Caltech101-20, Scene15, CCV, COIL100, Fashion, Hdigit, VGGFace2, and MNIST. The detailed descriptions of these datasets are listed in Table 1.

Baseline Algorithms. We benchmark the proposed OSAA-MVC against eleven representative large-scale MVC algorithms, including FMVACC [Wang *et al.*, 2022], 3AMVC [Ma *et al.*, 2024], AEVC [Liu *et al.*, 2024a], MV-CAGAF [Wang *et al.*, 2025], Orth-NTF [Li *et al.*, 2023], TC-MVSC [Chang *et al.*, 2024], SMVAGC [Wang *et al.*, 2024], MVSC-HFD [Ou *et al.*, 2024], NpGC [Yu *et al.*, 2024], MCHBG [Zhao *et al.*, 2025], LASD [Zhang *et al.*, 2025]. To ensure an equitable comparison, all selected competitors feature linear computational complexity with respect to the number of samples. It is worth mentioning that the first four methods address the Anchor-Unaligned Problem (AUP) by incorporating anchor alignment modules. The experimental environment consists of a 1.40 GHz Intel Core Ultra 7 CPU and 32GB RAM.

Experimental Details. To ensure a fair comparison with baselines, we optimize their hyperparameters by traversing the search spaces recommended in their papers and report the best results. Our method involves three key parameters: anchor number m, balance factor α, and regularization parameter λ. We adopt a grid search strategy to determine the optimal values. Specifically, m is selected from $\{k,2k,3k\}$, α is tuned within the range of $[0.1,0.5]$ with a step size of 0.2, and λ is searched from $\{10,100,1000\}$. Four clustering evaluation metrics are employed in our paper, including accuracy (ACC), normalized mutual information (NMI), Purity, and F-score. To mitigate the impact of random initialization, we repeat the clustering process 50 times and report the average results.

4.2 Clustering Performance

To evaluate the effectiveness of the proposed method, we compare OSAA-MVC with eleven state-of-the-art multi-view clustering algorithms on ten benchmark datasets, as shown in Table 2. Based on the experimental results, we have the following observations:

- OSAA-MVC achieves the best performance on all ten datasets across four evaluation metrics. Focusing on ACC, our method consistently surpasses the second-best method, with relative improvements of 6.05%, 4.06%, 23.29%, 10.43%, 0.27%, 2.42%, 3.87%, 0.74%, 6.60%, and 0.20%.
- Compared with methods that address the AUP, such as FMVACC, AEVC, and 3AMVC, our method achieves better clustering performance across datasets. The performance improvement validates the effectiveness of our one-step self-alignment strategy. Furthermore, the superior results confirm that our direct alignment paradigm outperforms indirect alignment by anchor graphs, avoiding the dependency and instability associated with baseline view selection.
- OSAA-MVC exhibits remarkable scalability and memory efficiency on large-scale datasets like Hdigit and MNIST. It delivers the best performance on large-scale benchmarks while several baselines suffer from out-of-memory errors, highlighting the suitability for large-scale applications.

4.3 Running Time

To evaluate computational efficiency, we report the average execution times of all compared methods on ten benchmark datasets, as summarized in Table 3. Specifically, on small-scale datasets, OSAA-MVC maintains competitive efficiency comparable to other efficient baselines. However, its advantages become pronounced on large-scale datasets. The proposed method achieves the fastest runtime on Scene15, CCV, COIL100, Fashion, VGGFace2, and MNIST. Furthermore, while competitors like Orth-NTF, MCHBG, and TC-MVSC suffer from out-of-memory errors on large datasets,

Dataset	FMVACC (NeurIPS'22)	AEVC (CVPR'24)	MV-CAGAF (TKDE'25)	3AMVC (MM'24)	MVSC-HFD (IF'24)	LASD (MM'25)	NpGC (AAAI'24)	SMVAGC (TIP'24)	Orth-NTF (NeurIPS'23)	MCHBG (IF'25)	TC-MVSC (TPAMI'24)	Proposed
						ACC(%)						
Yale	64.01	65.99	58.00	59.04	50.30	70.30	69.70	64.10	64.24	48.48	60.61	**74.55**
Dermatology	84.39	93.42	86.23	65.22	87.71	93.30	92.46	78.40	46.37	72.35	91.34	**97.21**
Caltech101-20	42.13	42.93	50.30	41.81	52.98	54.53	48.91	38.19	51.76	38.98	46.90	**67.23**
Scene15	40.02	42.11	28.07	34.35	33.71	46.42	39.78	38.79	44.59	29.72	29.72	**51.26**
CCV	19.19	24.07	16.92	17.47	20.48	23.30	25.90	23.44	15.40	12.31	15.92	**25.97**
COIL100	62.61	75.78	29.14	62.23	43.99	81.42	86.68	68.75	39.83	35.47	46.68	**88.78**
Fashion	79.26	77.92	73.19	75.45	70.18	81.84	77.76	78.67	30.15	32.69	57.58	**85.01**
Hdigit	86.55	89.68	73.77	83.52	73.61	91.00	84.69	66.99	77.39	40.87	OM	**91.67**
VGGFace2	10.97	11.65	6.88	10.54	10.80	13.11	14.67	14.84	9.05	6.25	OM	**15.82**
MNIST	95.60	95.02	97.59	97.92	98.02	97.28	98.84	87.04	OM	OM	OM	**99.04**
						NMI(%)						
Yale	67.85	69.35	64.75	64.63	52.16	69.54	72.35	69.75	73.34	54.64	62.42	**75.84**
Dermatology	76.27	88.98	84.50	62.53	80.42	86.90	85.23	68.13	26.63	64.36	85.18	**93.32**
Caltech101-20	57.05	56.04	55.77	47.21	52.14	60.88	60.82	44.44	58.74	33.57	57.71	**65.65**
Scene15	39.01	41.03	26.76	33.13	34.12	40.27	37.56	37.99	47.16	28.90	24.69	**48.04**
CCV	15.05	18.10	14.13	12.93	15.28	18.20	19.74	19.21	9.02	3.04	11.35	**20.30**
COIL100	82.56	87.99	53.38	82.12	72.58	93.04	95.63	84.99	67.86	38.93	66.18	**96.15**
Fashion	76.32	70.07	76.33	74.44	77.46	78.69	74.25	75.38	17.41	28.42	59.99	**79.86**
Hdigit	76.96	78.36	66.20	77.44	65.56	80.80	73.30	60.31	78.56	46.43	OM	**83.49**
VGGFace2	11.71	14.13	8.98	11.94	13.43	15.96	17.76	18.50	9.01	5.72	OM	**18.86**
MNIST	93.09	90.71	95.91	95.78	94.65	93.43	96.44	87.16	OM	OM	OM	**97.00**
						Purity(%)						
Yale	65.30	66.50	59.52	60.58	50.30	71.28	70.30	65.54	64.85	49.09	60.61	**75.76**
Dermatology	85.38	94.28	90.14	71.51	87.71	93.30	92.46	81.97	50.28	72.63	91.34	**97.21**
Caltech101-20	74.70	72.88	72.87	64.70	67.85	72.09	75.73	65.79	72.21	52.81	71.00	**76.32**
Scene15	43.13	46.03	31.35	37.30	36.03	46.71	44.08	43.40	49.72	34.60	30.57	**55.47**
CCV	23.08	26.30	21.08	21.81	23.79	25.97	28.45	26.46	18.80	12.83	18.20	**29.13**
COIL100	64.77	79.30	30.89	65.00	45.36	82.17	89.54	72.62	42.42	36.71	53.60	**91.01**
Fashion	80.23	78.09	77.16	78.14	70.97	81.84	78.22	80.71	30.90	35.51	57.58	**85.01**
Hdigit	86.59	89.77	74.67	84.30	73.61	91.00	84.69	70.04	77.39	49.48	OM	**91.67**
VGGFace2	11.95	12.83	7.33	11.15	11.27	13.99	15.67	15.87	10.23	6.43	OM	**16.93**
MNIST	96.33	95.43	97.82	98.12	98.02	97.28	98.84	87.93	OM	OM	OM	**99.04**
						F-score(%)						
Yale	50.46	52.25	44.37	42.93	31.64	51.99	56.28	51.59	49.82	33.58	41.47	**60.68**
Dermatology	78.01	90.62	83.63	57.99	81.98	89.95	86.95	68.95	35.57	64.62	86.98	**95.44**
Caltech101-20	35.24	34.89	46.17	35.87	47.62	46.80	40.93	33.68	45.40	23.63	31.32	**70.20**
Scene15	27.55	29.91	18.48	23.13	26.61	31.88	27.96	25.99	33.87	21.02	17.51	**37.78**
CCV	11.43	13.08	11.06	10.42	12.15	13.92	14.14	13.62	8.59	10.95	10.35	**14.50**
COIL100	56.97	70.05	20.70	57.83	37.24	78.25	85.81	61.84	28.54	3.28	11.46	**87.91**
Fashion	72.49	67.90	67.94	68.13	68.79	75.63	71.20	71.66	19.55	24.32	45.73	**77.32**
Hdigit	77.54	80.86	58.47	73.87	61.00	82.99	73.62	56.82	69.20	33.14	OM	**84.13**
VGGFace2	4.98	5.77	4.58	5.40	5.63	6.43	7.27	4.10	7.60	3.86	4.20	**7.81**
MNIST	93.98	91.89	96.50	96.69	96.12	94.64	97.68	83.75	OM	OM	OM	**98.09**

Table 2: Clustering performance comparison of the proposed method and eleven large-scale MVC methods on ten multi-view datasets. The best and second-best results are marked in bold and underline, respectively. 'OM' indicates the "out-of-memory error".

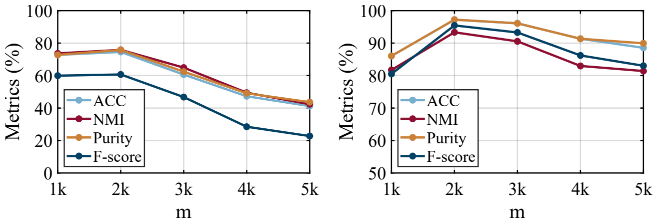
Dataset	FMVACC	AEVC	MV-CAGAF	3AMVC	MVSC-HFD	LASD	NpGC	SMVAGC	Orth-NTF	MCHBG	TC-MVSC	Proposed
Yale	3.77	1.06	4.89	5.77	180.65	4.38	3.08	8.79	0.66	**0.33**	0.87	3.93
Dermatology	**1.08**	3.46	2.07	5.63	3.13	3.28	4.69	1.21	4.39	9.64	2.23	1.71
Caltech101-20	15.05	21.06	40.03	19.19	8.97	167.40	14.43	40.15	26.72	**8.34**	11.71	19.00
Scene15	17.05	16.23	26.71	16.88	8.86	54.58	37.34	18.69	26.70	72.24	78.63	**7.86**
CCV	56.80	59.84	40.63	47.30	24.64	189.74	75.86	40.21	48.23	163.57	313.90	**16.97**
COIL100	142.75	95.64	129.26	149.83	102.99	492.34	624.70	60.81	260.21	93.51	279.78	**47.10**
Fashion	78.68	53.37	57.53	50.49	133.29	96.32	87.65	110.05	63.48	513.07	68590.44	**44.96**
Hdigit	37.93	38.90	51.80	71.76	**24.99**	79.93	103.82	109.29	56.78	213.51	OM	31.88
VGGFace2	210.42	414.16	617.71	317.08	230.29	787.32	468.98	1164.61	347.69	7681.96	OM	**136.41**
MNIST	535.54	626.79	699.83	597.48	544.01	899.18	1297.68	1220.25	OM	OM	OM	**182.67**

Table 3: Running time comparison (seconds) on ten multi-view datasets. The shortest and second-shortest times are highlighted in bold and underline, respectively. 'OM' indicates the "out-of-memory error".

our method handles them efficiently, demonstrating favorable scalability in both time and memory.

4.4 Ablation Study

To comprehensively evaluate the contribution of each component within our proposed framework, particularly the mixed anchor strategy and the self-alignment mechanism, we conduct ablation studies by comparing OSAA-MVC with three variants: (1) $\mathbf{A}_i$: using only view-specific anchors; (2) $\mathbf{A}$: using only the shared consensus anchors; and (3) $\mathbf{A}_{mix}$: utilizing the mixed anchor strategy but removing the alignment constraint. The results reported in Table 4 show the performance differences among different variants. First, both $\mathbf{A}_i$ and $\mathbf{A}$ underperform compared to the mixed variants, indicating that relying solely on specific or consensus information is insufficient for complex multi-view data. Sec-



(a) Yale (b) Dermatology

Figure 3: The sensitivity analysis of anchor number m on Yale and Dermatology datasets.

Dataset	$\mathbf{A}_i$	$\mathbf{A}$	$\mathbf{A}_{\text{mix}}$	Proposed
Yale	66.67	59.39	72.12	**74.55**
Dermatology	88.83	86.31	93.85	**97.21**
Caltech101-20	60.39	56.29	65.84	**67.23**
Scene15	41.34	40.51	46.24	**51.26**
CCV	23.92	23.82	24.10	**25.97**
COIL100	78.89	80.36	83.58	**88.78**
Fashion	80.31	73.24	81.73	**85.01**
Hdigit	87.64	87.61	90.27	**91.67**
VGGFace2	12.27	13.17	13.44	**15.82**
MNIST	97.07	95.67	98.18	**99.04**

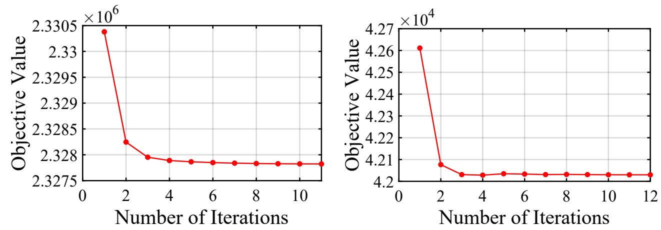
Table 4: Ablation study results in terms of ACC (%) on ten multi-view datasets. The best result is highlighted in bold.

ond, while $\mathbf{A}_{mix}$ improves upon the view-specific and consensus anchors, it is consistently outperformed by the proposed method. For instance, on the Dermatology dataset, the proposed method increases the clustering performance from 93.85% to 97.21% compared to $\mathbf{A}_{mix}$. This performance improvement validates the necessity of the self-alignment mechanism, demonstrating that simply mixing anchors without alignment introduces semantic mismatch, whereas our one-step alignment strategy effectively resolves the AUP.

4.5 Parameter Sensitivity

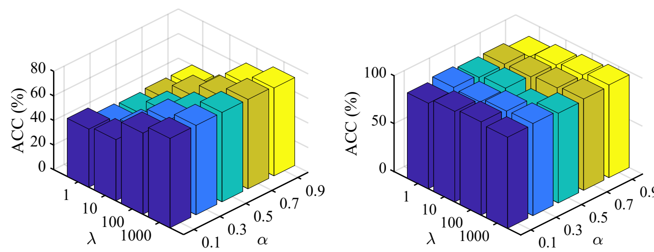
Sensitivity of anchor number m. We investigate the impact of the anchor number m on clustering performance by varying it from k to $5k$, as shown in Figure 3. From the results, the performance improves as m increases from k to $2k$ and reaches the best performance at $m = 2k$. However, further increasing m beyond $3k$ leads to a noticeable decline in clustering performance, indicating that too few anchors cannot sufficiently cover the data distribution, while too many anchors may introduce redundancy and degrade the consensus graph. Therefore, we set the search space of m to $\{k, 2k, 3k\}$ in the main experiments.

Sensitivity of parameters α and λ. To evaluate the stability of OSAA-MVC, we analyze the sensitivity of the balance factor α and the regularization parameter λ. As shown in Figure 5, we report the clustering performance in terms of ACC on the Yale and Dermatology datasets. The proposed OSAA-MVC is relatively insensitive to α but more sensitive to λ. Based on these empirical observations, we suggest selecting α from a coarse candidate set $\{0.1, 0.3, 0.5\}$ and tuning λ within the range of $\{10, 100, 1000\}$ for practical applications.



(a) Yale (b) Dermatology

Figure 4: The convergence analysis of our proposed method on Yale and Dermatology datasets.



(a) Yale (b) Dermatology

Figure 5: The sensitivity analysis of parameters α and λ on Yale and Dermatology datasets.

4.6 Convergence

To empirically investigate the convergence properties of the proposed optimization algorithm, we record the objective function values at each iteration on the Yale and Dermatology datasets. As illustrated in Figure 4, the objective value monotonically decreases as the number of iterations increases. Notably, the curve drops sharply within the first few iterations and stabilizes rapidly, typically reaching convergence in less than ten iterations. This observation verifies that our iterative optimization scheme is computationally efficient and capable of finding a local optimum quickly.

5 Conclusion

In this paper, we propose a novel unified one-step framework, named OSAA-MVC, to tackle the challenging Anchor-Unaligned Problem in large-scale multi-view clustering. Instead of traditional multi-stage approaches, OSAA-MVC integrates anchor generation, anchor alignment, and anchor graph construction into a coherent joint optimization process. By aligning the learned view-specific anchors with the learned global consensus anchors in the latent space, our method achieves direct self-alignment, effectively eliminating the dependency on the quality of initial anchor graphs. Furthermore, the proposed mixed anchor reconstruction strategy fuses common and view-specific information, successfully circumventing the dilemma of baseline view selection and the associated bias. Extensive experiments on various benchmark datasets validate that OSAA-MVC consistently outperforms state-of-the-art methods while maintaining high computational efficiency.

Acknowledgements

This work was supported in part by the Natural Science Foundation of Hunan Province, China, under Grant 2026JJ20079, and the National Natural Science Foundation of China (NSFC) under Grants 62376279 and 62306324.

Contribution Statement

* These authors contributed equally. † Corresponding author.

References

[Bezdek and Hathaway, 2003] James C Bezdek and Richard J Hathaway. Convergence of alternating optimization. *Neural, Parallel & Scientific Computations*, 11(4):351–368, 2003.

[Chang *et al.*, 2024] Wei Chang, Huimin Chen, Feiping Nie, Rong Wang, and Xuelong Li. Tensorized and compressed multi-view subspace clustering via structured constraint. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[Deng *et al.*, 2025] Xinyue Deng, Dongliang Wang, Xinfeng Li, Yuying Zhu, Yanan Li, and Zhenwen Ren. Multi-view clustering with tensor-aligned dynamic anchor graphs. *Neurocomputing*, page 132087, 2025.

[Dong *et al.*, 2023] Zhibin Dong, Siwei Wang, Jiaqi Jin, Xinwang Liu, and En Zhu. Cross-view topology based consistent and complementary information for deep multi-view clustering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19440–19451, 2023.

[Guan *et al.*, 2025] Renxiang Guan, Wenxuan Tu, Siwei Wang, Jiyuan Liu, Dayu Hu, Chang Tang, Yu Feng, Junhong Li, Baili Xiao, and Xinwang Liu. Structure-adaptive multi-view graph clustering for remote sensing data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 16933–16941, 2025.

[Hu *et al.*, 2022] Xingchen Hu, Xinwang Liu, Witold Pedrycz, Qing Liao, Yinghua Shen, Yan Li, and Siwei Wang. Multi-view fuzzy classification with subspace clustering and information granules. *IEEE Transactions on Knowledge and Data Engineering*, 35(11):11642–11655, 2022.

[Ji and Feng, 2023] Jintian Ji and Songhe Feng. Anchor structure regularization induced multi-view subspace clustering via enhanced tensor rank minimization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19343–19352, 2023.

[Ji and Feng, 2025] Jintian Ji and Songhe Feng. Anchors crash tensor: efficient and scalable tensorial multi-view subspace clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[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, pages 4412–4419, 2020.

[Li *et al.*, 2020] 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, 2020.

[Li *et al.*, 2023] Jing Li, Quanxue Gao, Qianqian Wang, Ming Yang, and Wei Xia. Orthogonal non-negative tensor factorization based multi-view clustering. *Advances in Neural Information Processing Systems*, 36:18186–18202, 2023.

[Liu *et al.*, 2024a] 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 *et al.*, 2024b] Suyuan Liu, Qing Liao, Siwei Wang, Xinwang Liu, and En Zhu. Robust and consistent anchor graph learning for multi-view clustering. *IEEE Transactions on Knowledge and Data Engineering*, 36(8):4207–4219, 2024.

[Ma *et al.*, 2024] Huimin Ma, Siwei Wang, Shengju Yu, Suyuan Liu, Jun-Jie Huang, Huijun Wu, Xinwang Liu, and En Zhu. Automatic and aligned anchor learning strategy for multi-view clustering. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 5045–5054, 2024.

[Ou *et al.*, 2024] Qiyuan Ou, Siwei Wang, Pei Zhang, Sihang Zhou, and En Zhu. Anchor-based multi-view subspace clustering with hierarchical feature descent. *Information Fusion*, 106:102225, 2024.

[Ren *et al.*, 2021] Zhenwen Ren, Xingfeng Li, Mithun Mukherjee, Yuqing Huang, Quansen Sun, and Zhen Huang. Robust multi-view graph clustering in latent energy-preserving embedding space. *Information Sciences*, 569:582–595, 2021.

[Shi *et al.*, 2021] Shaojun Shi, Feiping Nie, Rong Wang, and Xuelong Li. Fast multi-view clustering via prototype graph. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):443–455, 2021.

[Sun *et al.*, 2021] Mengjing Sun, Pei Zhang, Siwei Wang, Sihang Zhou, Wenxuan Tu, Xinwang Liu, En Zhu, and Changjian Wang. Scalable multi-view subspace clustering with unified anchors. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 3528–3536, 2021.

[Sun *et al.*, 2024] Yinghui Sun, Xingfeng Li, Quansen Sun, Min-Ling Zhang, and Zhenwen Ren. Improved weighted tensor schatten p-norm for fast multi-view graph clustering. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 1427–1436, 2024.

[Wang *et al.*, 2021] 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, 2021.

[Wang *et al.*, 2022] 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.*, 2024] Siwei Wang, Xinwang Liu, Suyuan Liu, Wenxuan Tu, and En Zhu. Scalable and structural multi-view graph clustering with adaptive anchor fusion. *IEEE Transactions on Image Processing*, 2024.

[Wang *et al.*, 2025] Siwei Wang, Xinwang Liu, Qing Liao, Yi Wen, En Zhu, and Kunlun He. Scalable multi-view graph clustering with cross-view corresponding anchor alignment. *IEEE Transactions on Knowledge and Data Engineering*, 2025.

[Wen *et al.*, 2019] Jie Wen, Zheng Zhang, Yong Xu, Bob Zhang, Lunke Fei, and Hong Liu. Unified embedding alignment with missing views inferring for incomplete multi-view clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5393–5400, 2019.

[Wen *et al.*, 2022] Jie Wen, Zheng Zhang, Lunke Fei, Bob Zhang, Yong Xu, Zhao Zhang, and Jinxing Li. A survey on incomplete multiview clustering. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 53(2):1136–1149, 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 31st 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.*, 2025] Jiaxuan Xu, Lei Duan, Xinye Wang, and Liang Du. Dynamic anchor-based ensemble clustering via hypergraph reconstruction. In *International Joint Conference on Artificial Intelligence*, 2025.

[Yang *et al.*, 2020] Ben Yang, Xuetao Zhang, Feiping Nie, Fei Wang, Weizhong Yu, and Rong Wang. Fast multi-view clustering via nonnegative and orthogonal factorization. *IEEE Transactions on Image Processing*, 30:2575–2586, 2020.

[Yu *et al.*, 2023] Weizhong Yu, Liyin Xing, Feiping Nie, and Xuelong Li. Multiview fuzzy clustering based on anchor graph. *IEEE Transactions on Fuzzy Systems*, 32(3):755–766, 2023.

[Yu *et al.*, 2024] Shengju Yu, Siwei Wang, Zhibin Dong, Wenxuan Tu, Suyuan Liu, Zhao Lv, Pan Li, Miao Wang, and En Zhu. A non-parametric graph clustering framework for multi-view data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 16558–16567, 2024.

[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, pages 16696–16704, 2024.

[Zhang *et al.*, 2025] Junpu Zhang, Shengju Yu, Suyuan Liu, Siwei Wang, Miaomiao Li, Xinwang Liu, En Zhu, and Kunlun He. Learning the anchors with similar distributions to original data for multi-view clustering. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 857–866, 2025.

[Zhao *et al.*, 2025] Zihua Zhao, Ting Wang, Haonan Xin, Rong Wang, and Feiping Nie. Multi-view clustering via high-order bipartite graph fusion. *Information Fusion*, 113:102630, 2025.

[Zhou *et al.*, 2025] Jie Zhou, Feiping Nie, Xinglong Luo, and Xingshi He. Hierarchical bipartite graph based multi-view subspace clustering. *Information Fusion*, 117:102821, 2025.