

# Anchor Learning with Potential Cluster Constraints for Multi-view Clustering

Yawei Chen<sup>1</sup>, Huibing Wang<sup>1\*</sup>, Jinjia Peng<sup>2</sup>, Yang Wang<sup>3\*</sup>

<sup>1</sup>School of Information Science and Technology, Dalian Maritime University, Dalian, China

<sup>2</sup>School of Cyber Security and Computer, Hebei University, Baoding, China

<sup>3</sup>School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, China  
{cyw,huibing.wang}@dlmu.edu.cn, pengjinjia@hbu.edu.cn, yangwang@hfut.edu.cn

## Abstract

Anchor-based multi-view clustering has received extensive attention due to its efficient performance. Existing methods only focus on how to dynamically learn anchors from the original data and simultaneously construct anchor graphs describing the relationships between samples and perform clustering, while ignoring the reality of anchors, i.e., high-quality anchors should be generated uniformly from different clusters of data rather than scattered outside the clusters. To deal with this problem, we propose a novel method termed Anchor Learning with Potential Cluster Constraints for Multi-view Clustering (ALPC) method. Specifically, ALPC first establishes a shared latent semantic module to constrain anchors to be generated from specific clusters, and subsequently, ALPC improves the representativeness and discriminability of anchors by adapting the anchor graph to capture the common clustering center of mass from samples and anchors, respectively. Finally, ALPC combines anchor learning and graph construction into a unified framework for collaborative learning and mutual optimization to improve the clustering performance. Extensive experiments demonstrate the effectiveness of our proposed method compared to some state-of-the-art MVC methods.

**Code** — <https://github.com/whbdmu/ALPC>

## Introduction

With the advent of the big data generation, there is an increasing diversity and complexity of data, which this data often exists in multiple forms (Ren et al. 2014; Cui et al. 2023; Wei et al. 2021). For example, news can be reported in a variety of formats, including text, images and videos. Due to these multiple sources of data, multi-view learning has attracted extensive attention (Huang et al. 2023; Wang et al. 2024a). Multi-view clustering is a fundamental task in multi-view learning, which tries to explore complementary information between views, discover the intrinsic structure of clusters, and group information from multiple sources (Qian et al. 2023; Huang et al. 2021; Wang et al. 2024b).

In previous investigations, researchers have contributed a vast number of Multi-view clustering (MVC) methods,

among which subspace learning has received extensive attention and examination as a particularly compelling research issue. MVC methods based on subspace learning usually learn the self-representation matrix and then perform spectral clustering. However, multi-view subspace clustering usually has high time complexity,  $\mathcal{O}(n^2)$  or even  $\mathcal{O}(n^3)$ . The high time complexity comes mainly from the construction of the self-representation matrix and the computation of the spectral embedding in the spectral clustering (Cui et al. 2024; Ling et al. 2023; Wang et al. 2016).

In order to address this problem, anchor-based multi-view clustering has been proposed in recent years, which aims to learn representative anchors from multiple samples, and then learn the anchor graph through the relationship between the samples and the anchors and thus perform clustering. Obviously, how to learn high-quality anchors is essential. Existing anchor learning strategies can be categorized into two ways: static selection and dynamic learning. The static selection strategies include random sampling and k-means, where random sampling aims to randomly select data from all the samples to be used as anchors, while k-means directly utilizes the sample center of mass as an anchor. Although the above methods are simple and effective in the process of selecting anchors, the result of selecting anchors is random, which leads to inferior quality of anchors, thus reducing the clustering performance. In order to improve the quality of anchors, dynamic anchor learning become popular. For example, (Wang et al. 2022a) integrates anchor selection and subspace graph construction in a unified framework so that the two processes negotiate with each other to improve the quality of the anchors, avoiding the need to select anchors directly from the original samples. (Sun et al. 2021a), on the other hand, adaptively learns the consensus anchor graph of all samples in the projection space to be used for clustering.

Although MVC methods for dynamic anchor learning achieve excellent performance, they only learn anchors dynamically through simple low-rank constraints, which may lead to uneven anchor distribution, i.e., some anchors will be scattered outside the data clusters or some data clusters lack corresponding anchors. That is, these methods ignore the reality of anchors, i.e., high-quality anchors should be naturally generated from different data clusters and share the underlying clustering structure with the original data, rather than being scattered outside the data clusters.

\*Corresponding Author.

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

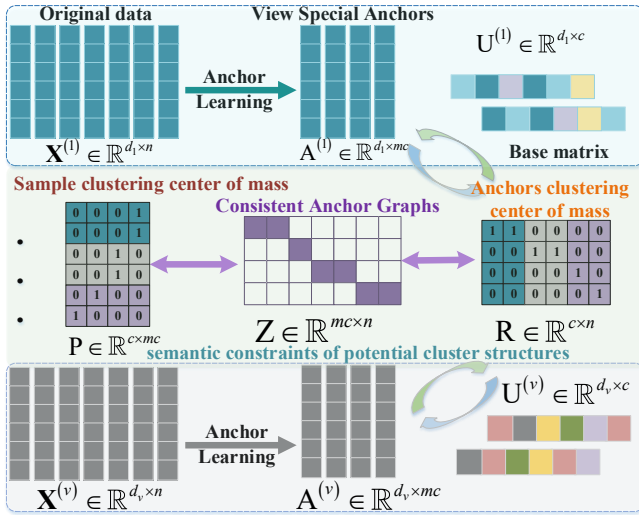


Figure 1: Schematic diagram of the proposed ALPC. Aims to learn the most representative anchors for large multi-view datasets. ALPC unifies anchor learning and consistent anchor graphs into a single model. ALPC adopts the semantic constraint of sharing potential cluster structures with the aim of guiding the anchors to be generated uniformly from different clusters of the original data. At the same time, in order to ensure the representative nature of the generated anchors, ALPC discretizes the anchor graphs to guide the center of mass clustering of the original data and the mass clustering centers of the anchors are aligned, thus constraining both to share the same clustering structure.

In order to solve the above problems, this paper proposes the ALPC method, the framework of which is shown in Fig. 1. Different from the existing multi-view clustering based on anchor graph, firstly, ALPC innovatively proposes the semantic constraint of sharing potential cluster structure, which aims to guide anchors to be generated uniformly in sample clusters, avoiding the situation that anchors are generated outside the clusters or there are no anchors in the sample clusters; secondly, ALPC performs the feature decomposition of the anchor graph to learn the common clustering center of gravity of the samples and anchors, aiming at matching the anchor clustering structure with the original sample clustering structure to match the anchors with a fairer, more discriminative and clearer clustering structure; finally, ALPC unifies anchor selection and potential cluster structure constraints in a single framework so that the two processes negotiate with each other to improve the quality of the anchors, and thus adaptively learns low-rank anchor graphs for clustering. The main contributions of this paper are totaled as follows when compared to existing MVC methods based anchors:

- Unlike existing dynamic anchor learning strategies, this paper innovatively proposes to share the semantic constraints of potential cluster structures to adaptively learn representative anchors uniformly from sample clusters.
- Learning the cluster structure of samples and anchors by

Notations	Descriptions
$n, l, c$	The number of samples, views and clusters
$m$	Number of anchors in each cluster
$d_v$	Feature dimension of the $v$ -th view
$\mathbf{X}^{(v)} \in \mathbb{R}^{d_v \times n}$	Data matrix in the $v$ -th view
$\mathbf{A}^{(v)} \in \mathbb{R}^{d_v \times mc}$	Anchor matrix of the $v$ -th view
$\mathbf{Z} \in \mathbb{R}^{mc \times n}$	Consensus subspace representation
$\mathbf{U}^{(v)} \in \mathbb{R}^{d_v \times c}$	Anchored cluster center of mass
$\mathbf{P} \in \mathbb{R}^{c \times mc}$	Clustering Indicator for anchors
$\mathbf{R} \in \mathbb{R}^{c \times n}$	Clustering Indicator for original data

Table 1: The Descriptions of Significant Formula Notations

fitting the relevance of anchors and learning the cluster structure with differentiation, clarity, and fair anchoring cluster structure by identifying the clustering center of mass of samples and anchors through anchor graphs.

- Extensive experiments on six benchmark datasets demonstrate the effectiveness of our method compared to state-of-the-art methods.

## Related Work

In this section, we will present existing work related to the research in this paper, including methods for multi-view subspace clustering and anchor graph-based multi-view clustering. Table 1 lists the main notations used in this paper.

## Multi-view Subspace Clustering

In recent years, multi-view subspace clustering methods have gained extensive attention due to their low-rank representations that can exploit the intrinsic subspace structure of the data to obtain good clustering performance. Most of these can be described in the following framework:

$$\min_{\mathbf{S}^{(v)}, \mathbf{S}} \sum_{v=1}^l \underbrace{\left\| \mathbf{X}^{(v)} - \mathbf{X}^{(v)} \mathbf{S}^{(v)} \right\|_F^2}_{\text{Graph Construction}} + \underbrace{\Omega(\mathbf{S}^{(v)}, \mathbf{S})}_{\text{Fusion}} \quad (1)$$

$s.t. \text{diag}(\mathbf{S}^{(v)}) = 0, \text{diag}(\mathbf{S}) = 0, \mathbf{S}_i^T \mathbf{1} = 1.$

where  $\mathbf{S}^{(v)} \in \mathbb{R}^{n \times n}$  denotes the subspace representation of the  $v$ -th view and  $\Omega$  denotes the uniform consensus regularization term with the aim of obtaining the fusion graph  $\mathbf{S} \in \mathbb{R}^{n \times n}$ .

Most existing multi-view subspace clustering methods have evolved from the aforementioned framework because they can capture global structures and obtain low-rank graphs. For example, (Cao et al. 2015) proposed inducing the Hilbert Schmidt Independence Criterion (HISC) to learn complementary information from different views, and obtained clustering results by imposing low-rank constraints on the learned affinity graph. (Luo et al. 2018; Wang 2021; Wang et al. 2022b) divided the subspace representation of each view into a fixed part and a specific part to retain consistency and complementary information between views, making the subspace representation more precise. However, these methods ignore high-order information between affinity graphs, so recently, methods based on low-rank tensor

constraints have become popular. These methods stack different views' affinity graphs into a tensor to explore high-order information between views, such as (Xia et al. 2023; Jiang et al. 2022; Lv et al. 2023).

The above methods have good performance in clustering. However, these methods cannot avoid constructing a complete  $n \times n$  graph, which leads to a time complexity of  $\mathcal{O}(n^3)$ , making them powerless in large-scale data sets and limiting their application in real scenarios.

### Anchor-based Multi-view Clustering

In the field of multi-view subspace clustering, anchor graphs have been recognized as an effective means of dealing with large-scale datasets. The principle of the anchor graph approach is to select  $m$  representative anchors among  $n$  samples and explore the relationship between the anchors and the samples. As a result, the construction of the anchor graph is reduced from  $\mathbf{S} \in \mathbb{R}^{n \times n}$  to  $\mathbf{Z} \in \mathbb{R}^{m \times n}$ , which undoubtedly greatly reduces the computation time as well as storage and maintains the excellent clustering performance.

Recently, a representative of the anchor-based MVC approach, (Wang et al. 2022a) proposed a fast and parameter-free multi-view clustering method, i.e., learning anchors and constructing anchor graphs in a unified framework. (Liu et al. 2024) constructs a consistent anchor graph and divides the anchor graph into a specific portion as well as a noise portion, and then imposes low-rank constraints on the specific portion. In addition, (Li et al. 2022) proposes an alternative anchor sampling strategy to construct individual anchor graphs, which are then merged into the consensus graph.

Although these methods have achieved considerable performance, the existing work still has the following considerations that can be improved: (1) anchor selection is performed independently in each view and lacks mutual consultation with other views, leading to poor performance of the learned representation. (2) Existing methods simply learn anchors dynamically but ignore the underlying cluster structure shared between anchors and raw data, resulting in poor-quality anchors. In the next section, this paper proposes a new MVC approach to address these issues.

## The Proposed Method

In this section, the ALPC method proposed in this paper is described in detail and then an alternating optimization algorithm is provided to solve the resulting problem. This will be followed by a discussion of the time complexity as well as the convergence of ALPC.

### Formulation

For anchors and graph learning, each view can be represented as a linear concatenation of all anchors. Based on this assumption, we obtain view-specific anchors and consensus subspace representation:

$$\begin{aligned} \min_{\mathbf{A}^{(v)}, z_i} \sum_{v=1}^l \sum_{i=1}^n \left\| x_i^{(v)} - \mathbf{A}^{(v)} z_i \right\|_2^2 \\ \text{s.t. } \mathbf{A}^{(v)T} \mathbf{A}^{(v)} = \mathbf{I}, z_i \geq 0, z_i^T \mathbf{1} = 1. \end{aligned} \quad (2)$$

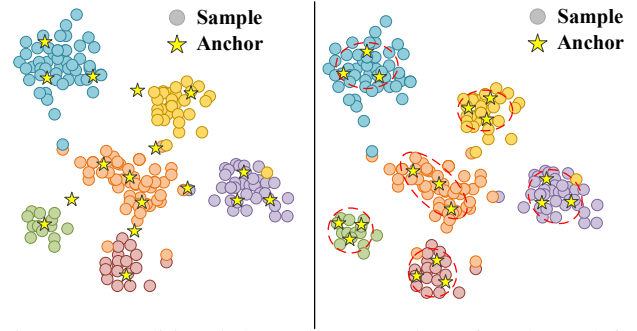


Figure 2: Traditional dynamic generation of anchors (left) and our proposed cluster-shared generation of anchors (right)

where  $x_i^{(v)}$  and  $z_i$  are the  $i$ -th columns of  $\mathbf{X}^{(v)}$  and  $\mathbf{Z}$ , denoting the corresponding data and linear coefficients. Applying orthogonal constraints to  $\mathbf{A}^{(v)}$  makes the learned anchors more diverse.

However, a simple orthogonality constraint does not guarantee that the generated anchors are representative, and such a constraint may result in some complex data not being represented. Fortunately, we can constrain the learning of anchors based on the real-world significance of anchors, i.e., representative anchors should be generated uniformly from different data clusters, as shown in Figure 2.

Among them, in Figure 2, the left side is the traditional dynamic anchor learning strategy, and it can be seen that the anchors learned by the traditional method are not only unevenly distributed, some clusters contain a small number of anchors, and some clusters contain a large number of anchors, and some of the anchors are outside the clusters, and it is obvious that these outside anchors are not representative, because it may represent different cluster information. On the right is the new strategy of anchor learning proposed in this paper, i.e., generating anchors from clusters dynamically, so that the learned anchors are not only representative but also evenly distributed among different clusters. Inspired by this, we make the following constraints to learn more representative anchors:

$$\begin{aligned} \min_{\Phi} \sum_{v=1}^l \lambda_1 \left\| \mathbf{A}^{(v)} - \mathbf{U}^{(v)} \mathbf{P} \right\|_F^2 + \lambda_2 \left\| \mathbf{Z} - \mathbf{P}^T \mathbf{R} \right\|_F^2 \\ \text{s.t. } \mathbf{U}^{(v)T} \mathbf{U}^{(v)} = \mathbf{I}, \mathbf{A}^{(v)T} \mathbf{A}^{(v)} = \mathbf{I} \end{aligned} \quad (3)$$

where  $\mathbf{U}^{(v)}$  is a basis matrix on which orthogonal constraint are imposed to avoid infinite solutions,  $\mathbf{P}$  denotes the potential clustering indicators of the anchors, and  $\mathbf{R}$  denotes the clustering indicators of the original data. In order to align the clustering center of mass of anchors with that of the original data, ALPC obtains the clustering center of mass of the original data as well as the clustering center of anchors by decomposing the anchor graph, thus constraining the two to share the same clustering structure and thus aligning the clustering centers of the two. Combining Eq. (2) and Eq. (3), we express the objective function of ALPC as follows:

$$\begin{aligned}
& \min_{\Phi} \sum_{v=1}^l \sum_i^n \left\| x_i^{(v)} - \mathbf{A}^{(v)} z_i \right\|_2^2 + \\
& \sum_{v=1}^l \lambda_1 \left\| \mathbf{A}^{(v)} - \mathbf{U}^{(v)} \mathbf{P} \right\|_F^2 + \lambda_2 \left\| \mathbf{Z} - \mathbf{P}^T \mathbf{R} \right\|_F^2 \\
& \text{s.t. } \mathbf{U}^{(v)T} \mathbf{U}^{(v)} = \mathbf{I}, \mathbf{A}^{(v)T} \mathbf{A}^{(v)} = \mathbf{I}, z_i \geq 0, z_i^T \mathbf{1} = 1.
\end{aligned} \tag{4}$$

where  $\lambda_1$  and  $\lambda_2$  are equilibrium parameters. Eq. (4) combines anchor graph learning and dynamic anchor learning into a unified framework. The first term is to learn anchors with diversity as well as anchor graphs, while the second term is to learn the underlying clustering structure of the anchors, and the third term is to align the anchors with the clustering centers of mass of the original data so that the anchors can be generated from the clustered clusters.

### Optimization

The optimization problem in Eq. (4) is not jointly convex when considering all variables simultaneously. Hence, we propose an alternating algorithm to optimize each variable while the others maintain fixed.

**Update  $\mathbf{A}^{(v)}$**  Since different  $\mathbf{A}^{(v)}$  is independent of each other on different views, it is possible to solve each  $\mathbf{A}^{(v)}$  independently. Fixing the other variables, the subproblem w.r.t.  $\mathbf{A}^{(v)}$  is:

$$\begin{aligned}
& \min_{\mathbf{A}^{(v)}} \sum_{i=1}^n \left\| x_i^{(v)} - \mathbf{A}^{(v)} z_i \right\|_2^2 + \lambda_1 \left\| \mathbf{A}^{(v)} - \mathbf{U}^{(v)} \mathbf{P} \right\|_F^2 \\
& \text{s.t. } \mathbf{A}^{(v)T} \mathbf{A}^{(v)} = \mathbf{I}
\end{aligned} \tag{5}$$

Setting the derivative of  $\mathbf{A}^{(v)}$  to zero solves the above problem:

$$\mathbf{A}^{(v)} \mathbf{Z} \mathbf{Z}^T + \lambda_1 \mathbf{A}^{(v)} - \lambda_1 \mathbf{U}^{(v)} \mathbf{P} - \mathbf{X}^{(v)} \mathbf{Z}^T = \mathbf{0} \tag{6}$$

Obviously, the optimal solution is:

$$\mathbf{A}^{(v)} = \left( \lambda_1 \mathbf{U}^{(v)} \mathbf{P} + \mathbf{X}^{(v)} \mathbf{Z}^T \right) \left( \mathbf{Z} \mathbf{Z}^T + \lambda_1 \mathbf{I} \right)^{-1} \tag{7}$$

**Update  $\mathbf{Z}$**  With fixing other variables, the sub-problem w.r.t.  $\mathbf{Z}$  is:

$$\min_{\mathbf{Z}} \sum_{v=1}^l \sum_{i=1}^n \left\| x_i^{(v)} - \mathbf{A}^{(v)} z_i \right\|_2^2 + \lambda_2 \left\| \mathbf{Z} - \mathbf{P}^T \mathbf{R} \right\|_F^2 \tag{8}$$

Similar to solving  $\mathbf{A}^{(v)}$ , the optimal solution can be obtained by setting the derivative of  $\mathbf{Z}$  to zeros:

$$\mathbf{Z} = \left( \sum_{v=1}^l \mathbf{A}^{(v)T} \mathbf{A}^{(v)} + \lambda_2 \mathbf{I} \right)^{-1} \left( \sum_{v=1}^l \mathbf{X}^{(v)T} \mathbf{X}^{(v)} + \lambda_2 \mathbf{P}^T \mathbf{R} \right) \tag{9}$$

**Update  $\mathbf{P}$**  Fixing other variables optimization problem w.r.t.  $\mathbf{P}$  is:

$$\min_{\mathbf{P}} \sum_{v=1}^l \left( \lambda_1 \left\| \mathbf{A}^{(v)} - \mathbf{U}^{(v)} \mathbf{P} \right\|_F^2 \right) + \lambda_2 \left\| \mathbf{Z} - \mathbf{P}^T \mathbf{R} \right\|_F^2 \tag{10}$$

---

### Algorithm 1: ALPC algorithm

---

**Input:** Multi-view data  $\{\mathbf{X}^{(v)}\}_{v=1}^l$ , cluster number  $k$ , parameters  $\lambda_1, \lambda_2, m$ ;

**Initialize:**  $\mathbf{A}^{(v)} = \mathbf{0}, \mathbf{U}^{(v)} = \mathbf{0}, \mathbf{P} = \mathbf{0}, \mathbf{R} = \mathbf{0}$

**Output:** Perform k-means on  $\mathbf{Z}$  to obtain the clusters.

- 1: **while** not converged **do**
  - 2:   Update variable  $\mathbf{A}^{(v)}$  using Eq. (7);
  - 3:   Update variable  $\mathbf{Z}$  using Eq. (9);
  - 4:   Update variable  $\mathbf{P}$  using Eq. (11);
  - 5:   Update variable  $\mathbf{R}$  using Eq. (13);
  - 6:   Update variable  $\mathbf{U}^{(v)}$  using Eq. (15);
  - 7: **end while**
- 

This can be solved by solving the following problem:

$$\mathbf{P} = \left( \lambda_1 \mathbf{I} + \lambda_2 \mathbf{R}^T \mathbf{R} \right)^{-1} \left( \lambda_1 \sum_{v=1}^l \mathbf{U}^{(v)T} \mathbf{A}^{(v)} + \lambda_2 \mathbf{R} \mathbf{Z}^T \right) \tag{11}$$

**Update  $\mathbf{R}$**  Fixing other variables optimization problem w.r.t.  $\mathbf{R}$  is:

$$\min_{\mathbf{R}} \lambda_2 \left\| \mathbf{Z} - \mathbf{P}^T \mathbf{R} \right\|_F^2 \tag{12}$$

This can be solved by solving the following problem:

$$\mathbf{R} = \left( \mathbf{P} \mathbf{P}^T \right)^{-1} \mathbf{P} \mathbf{Z} \tag{13}$$

**Update  $\mathbf{U}^{(v)}$**  Fixing other variables optimization problem w.r.t.  $\mathbf{U}^{(v)}$  is:

$$\min_{\mathbf{U}^{(v)}} \left\| \mathbf{A}^{(v)} - \mathbf{U}^{(v)} \mathbf{P} \right\|_F^2 \quad \text{s.t. } \mathbf{U}^{(v)T} \mathbf{U}^{(v)} = \mathbf{I} \tag{14}$$

It is equivalent to solving:

$$\max_{\mathbf{U}^{(v)}} \text{Tr} \left( \mathbf{U}^{(v)} \mathbf{A}^{(v)} \mathbf{P}^T \right) \quad \text{s.t. } \mathbf{U}^{(v)T} \mathbf{U}^{(v)} = \mathbf{I} \tag{15}$$

Since this is an orthogonal problem, it can be computed by singular value decomposition (SVD) of  $\mathbf{A}^{(v)} \mathbf{P}^T$ , i.e.,  $\mathbf{A}^{(v)} \mathbf{P}^T = \mathbf{U} \Sigma \mathbf{V}^T$ . The optimal solution is  $\mathbf{U}^{(v)} = \mathbf{U} \mathbf{V}^T$ .

From the above optimization, it can be seen that ALPC needs to update five variables, and each subproblem has a corresponding solution. For convenience, the whole optimization process is summarized as Algorithm 1.

### Complexity and Convergence Analysis

In this section, we will analyze the computational complexity of ALPC. It includes space complexity, time complexity and convergence.

**Memory complexity:** ALPC needs to store the matrices  $\mathbf{Z} \in \mathbb{R}^{mc \times n}$ ,  $\mathbf{A}^{(v)} \in \mathbb{R}^{d_v \times mc}$ ,  $\mathbf{P} \in \mathbb{R}^{c \times mc}$ . So the main space cost of ALPC is  $\mathcal{O}(ndv + nmv)$ , which proves its linear space complexity. **Time complexity:** The time consumption consists mainly of the update of five variables. For the update of the variable  $\mathbf{Z}$ , most of the time consumed is to find the product of the matrices and the inverse of the matrices, with a time complexity of  $\mathcal{O}(m^3 c^3 + m^2 c^2 d + m^2 c^2 n + mcdn)$ . Optimize the time

Dataset	Metrics	LMVSC	SMVSC	OPMC	OMSC	SFMC	FPMVS	EOMSC	AWMVC	FDAGF	Ours
MSRC	ACC	0.3476	0.6952	0.6286	0.7048	0.7286	0.6047	0.5905	<u>0.7810</u>	0.6429	<b>0.8819</b>
	NMI	0.2461	0.6190	0.5533	0.6314	0.6987	0.5557	0.4352	<u>0.7160</u>	0.6270	<b>0.7758</b>
	Purity	0.4048	0.6952	0.6381	0.7381	0.7667	0.6190	0.6048	<u>0.7810</u>	0.7571	<b>0.8819</b>
	F-score	0.2480	0.5766	0.5125	0.6123	0.6736	0.5029	0.4261	<u>0.6852</u>	0.5537	<b>0.7637</b>
BBCSport	ACC	0.6544	0.5091	0.6691	0.4522	<u>0.8254</u>	0.4209	0.4154	0.6397	0.4669	<b>0.9512</b>
	NMI	0.4471	0.2118	0.6705	0.2129	<u>0.7770</u>	0.1508	0.2367	0.4820	0.3749	<b>0.8410</b>
	Purity	0.6636	0.5367	0.7445	0.5110	0.8566	0.5183	0.5037	0.6890	<u>0.9228</u>	<b>0.9512</b>
	F-score	0.4854	0.3645	0.6501	0.3556	<u>0.7829</u>	0.3247	0.3468	0.5234	0.4978	<b>0.9009</b>
Wiki	ACC	0.1884	0.3273	0.1766	0.3535	0.3039	0.3140	<u>0.5415</u>	0.2170	0.5265	<b>0.6040</b>
	NMI	0.0469	0.1628	0.0454	0.1991	0.3196	0.1715	<u>0.5290</u>	0.0783	0.5149	<b>0.5514</b>
	Purity	0.2083	0.3433	0.2069	0.3765	0.3458	0.3367	0.6141	0.2547	<b>0.6811</b>	0.6399
	F-score	0.1294	0.2103	0.1244	0.2243	0.2113	0.2146	<u>0.4832</u>	0.1432	0.4325	<b>0.5105</b>
Caltech101-all	ACC	0.2005	0.2750	0.2188	0.2512	0.1777	<u>0.3015</u>	0.2414	0.2367	0.2997	<b>0.3284</b>
	NMI	0.2145	<u>0.3510</u>	0.2611	0.2522	0.2613	0.3549	0.2154	0.2397	0.2679	<b>0.3697</b>
	Purity	0.2396	0.3395	0.2691	0.2512	0.2854	<u>0.3460</u>	0.2460	0.2691	0.2619	<b>0.3617</b>
	F-score	0.1586	0.1462	0.0691	0.1422	0.0462	<u>0.2326</u>	0.1154	0.0697	0.1669	<b>0.2431</b>
MNIST	ACC	0.9896	0.9884	0.8392	0.8922	<u>0.9905</u>	0.9884	0.9808	0.9885	0.9887	<b>1.0000</b>
	NMI	0.9685	0.9650	0.9134	0.9336	<u>0.9709</u>	0.9650	0.9475	0.9647	0.9658	<b>0.9879</b>
	Purity	<u>0.9896</u>	0.9884	0.9385	0.8922	0.9813	0.9768	0.9621	0.9761	0.9735	<b>1.0000</b>
	F-score	0.9794	0.9768	0.8550	0.8923	<u>0.9813</u>	0.9768	0.9621	0.9671	0.9735	<b>0.9981</b>
YouTubeFace_sel	ACC	0.1479	0.1749	0.1556	0.1897	0.2104	<u>0.2414</u>	0.1924	0.2367	0.2146	<b>0.3046</b>
	NMI	0.1327	0.1091	0.0961	0.1225	0.0593	0.1327	0.1154	<u>0.2097</u>	0.1966	<b>0.2536</b>
	Purity	0.1216	0.1433	0.1069	0.1365	0.0958	<u>0.3279</u>	0.1141	0.1547	0.0811	<b>0.3299</b>
	F-score	0.0849	0.0903	0.0244	0.1243	0.0713	<u>0.1433</u>	0.0832	0.1032	0.0732	<b>0.1532</b>

Table 2: The Clustering performance of the compared multi-view clustering methods, including ACC, NMI, Purity and F-score. the best of these results are highlighted in bold. Underlining indicates sub-optimal performance.

complexity of  $\mathbf{A}^{(v)}$  to  $\mathcal{O}(m^3c^3 + d_vcmn)$ . For updating  $\mathbf{P}$ , then the complexity of  $\mathcal{O}(nc^2d^2 + c^3d^3 + 2nmcd)$  is required. For updating  $\mathbf{R}$ , then the complexity of  $\mathcal{O}(c^3md)$  is required. For updating  $\mathbf{U}^{(v)}$ , then the complexity of  $\mathcal{O}(d_vmc^2)$  is required. For updating  $\mathbf{P}$ , then the complexity of  $\mathcal{O}(nc^2d^2 + c^3d^3 + 2nmcd)$  is required. The computational complexity of the proposed ALPC is linearly related to the number of samples  $n$  due to  $m, d, v \ll n$ . So overall time complexity is  $\mathcal{O}(n)$ .

**Convergence:** From Eq. (4), it can be seen that the whole function is not convex when all variables are considered simultaneously. So we use an alternating optimization algorithm to update the resolution for each variable.

Let  $\mathcal{J}(\mathbf{Z}_t, \mathbf{A}_t, \mathbf{P}_t, \mathbf{R}_t, \mathbf{U}_t)$  denote the value of the objective function at the  $t$ -th iteration, it can be obtained as  $\mathcal{J}(\mathbf{Z}_t, \mathbf{A}_t, \mathbf{P}_t, \mathbf{R}_t, \mathbf{U}_t) \leq \mathcal{J}(\mathbf{Z}_{t+1}, \mathbf{A}_t, \mathbf{P}_t, \mathbf{R}_t, \mathbf{U}_t) \leq \dots \leq \mathcal{J}(\mathbf{Z}_{t+1}, \mathbf{A}_{t+1}, \mathbf{P}_{t+1}, \mathbf{R}_{t+1}, \mathbf{U}_{t+1})$ . This shows that the objective function value of ALPC decreases monotonically. Since the lower bound value of the algorithm is zero, it can be obtained that the algorithm can converge to a minimum value.

## Experiment

In this section, we conduct extensive experiments to compare ALPC with state-of-the-art methods and analyze the performance of the algorithm.

## Experiment Settings

We validate the well-behaved performance of ALPC using six benchmark datasets, where the maximum number of samples used is more than 100,000. The datasets used in the experiments include **MSRC** (Chen et al. 2021), **BBCSport**<sup>1</sup>, **Wiki**<sup>2</sup>, **Caltech101**<sup>3</sup>, **MNIST**<sup>4</sup> and **YouTubeFace sel**<sup>5</sup>. The details of the different datasets, including the number of samples, the number of views, the number of categories, and the feature dimensions of the dataset, are labeled in Table 3.

Our proposed algorithm is compared with nine state-of-the-art methods and the comparative algorithms are summarized as follows: **LMVSC** (Kang et al. 2019), **SMVSC** (Sun et al. 2021b), **OPMC** (Liu et al. 2021), **OMSC** (Chen et al. 2022), **SFMC** (Li et al. 2022), **FPMVS** (Wang et al. 2022a), **EOMSC** (Liu et al. 2022), **AWMVC** (Wan et al. 2023) and **FDAGF** (Zhang et al. 2023). Among them, the dynamic anchor learning strategies include SMVSC, OPMC, OMSC, FPMVS, EOMSC, AWMVC, FDAGF, and our proposed ALPC, while the static anchor strategies include LMVSC as well as SFMC. In this section, the clustering accuracies of the two anchor learning strategies will be compared, as well as the performance of ALPC in many dy-

<sup>1</sup><http://mlg.ucd.ie/datasets/bbc.html>

<sup>2</sup><http://www.svcl.ucsd.edu/projects/crossmodal/>

<sup>3</sup><https://paperswithcode.com/dataset/caltech-101>

<sup>4</sup><https://yann.lecun.com/exdb/mnist/>

<sup>5</sup><http://www.cs.tau.ac.il/~wolf/ytfaces/>

Dataset	$n$	$v$	$c$	$d$
MSRC	210	4	7	24/512/256/254
BBCSport	544	2	5	3183/3203
Wiki	2866	2	10	128/10
Caltech101 – all	9144	5	102	48/40/254/512/928
MNIST	60000	3	10	342/1024/64
YouTubeFace_sel	101499	5	31	64/512/64/647/838

Table 3: Summary of the used MVC benchmark datasets, where  $n$ ,  $v$ ,  $c$ ,  $d$  denote the number of samples, number of views, number of clusters, and feature dimensions, respectively

dynamic anchor learning. For a fair comparison, we used the official codes of the corresponding methods and ran K-means 50 times to get the best results. The optimal parameters for these methods are the optimal parameters tuned by searching through the grid in the suggested range. For the methods in this paper, we adjust the values of  $\lambda_1$  and  $\lambda_2$  in  $\{10^{-2}, 10^{-1}, \dots, 10^2\}$ ,  $\{10^{-4}, \dots, 10^{-1}, 1\}$ . In this paper, four popular evaluation metrics are taken including Accuracy (ACC), Normalized Mutual Information (NMI), Purity and F-score. For all the metrics, higher values indicate superior method performance.

## Clustering Results

Our comparison of ALPC and state-of-the-art methods was performed on six benchmark datasets on four metrics. Among them, Table 3 presents the results for ACC, NMI, Purity and F-score. From Table 3, the following conclusions are clearly observed:

- ALPC outperforms all other methods on most datasets. For example, the ACC of ALPC on MSRC, BBCsport, and MNIST are 10.09%, 12.58%, and 0.95% higher than the second-best method, respectively, which demonstrates the effectiveness of our method.
- Compared to LMVSC, better performance can be obtained by methods such as SMVSC, FPMVS, EOMSC and our proposed ALPC. The reason is that the former uses a static anchor selection strategy, while the latter all dynamically learn anchors and obtain an anchor graph, which makes the learned anchors more representative and discriminative, resulting in better clustering performance.
- Most methods do so without considering that anchors should be uniformly distributed in each data cluster, and the quality of the learned anchors is poor. The learned anchor graphs are discriminative because ALPC introduces the shared potential consistency semantic constraints to guide the anchors to be generated uniformly and stably in each data cluster, which makes the anchors more representative and discriminative. This gives ALPC better and more stable clustering results.

## Parameter Analysis

In this section, we use the grid search to tune two parameters  $\lambda_1$  and  $\lambda_2$  in ALPC in the ranges  $\{10^{-4}, \dots, 10^{-1}, 1\}$ ,

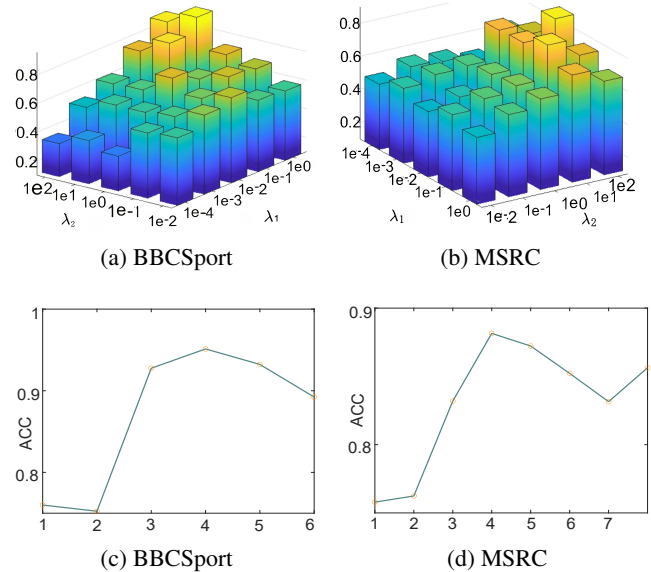


Figure 3: (a) and (b) are sensitivity analyses of hyperparameters  $\lambda_1$  and  $\lambda_2$  to the test benchmark dataset, while (c) and (d) are sensitivity analyses of the number of anchors  $m$  on two different datasets.

$\{10^{-2}, 10^{-1}, \dots, 10^2\}$  thereby investigating the effect of the parameters on ALPC.

Among them, we conducted parameter sensitivity experiments on two benchmark datasets and recorded the clustering performance of ALPC for different combinations of the two parameters. Figure 3 shows the ACC values of ALPC for different combinations of parameters. From the figure, we can get the following conclusions: (1) When the values of parameters  $\lambda_1$  and  $\lambda_2$  are located at  $\{10^{-1}, 1, 10\}$ , ALPC can get the best performance. (2) When parameter  $\lambda_2$  is too small, the results drop very fast, which indicates that ALPC is more sensitive to parameter  $\lambda_2$ . In addition, we also find that the number of anchors has a very large impact on the clustering performance. Obviously, most datasets perform best when the number of anchors is  $2c$  or  $3c$ . This proves the importance of anchor selection.

## Qualified Study

To demonstrate the superiority of the consensus graph learned by ALPC, we visualized the consensus graph of APLC on two benchmark datasets, as shown in Figure 4. In the figure, we show the consensus graphs learned by APLC on **BBCSport** and **MSRC** dataset. A clear block structure corresponding to the clustering of a large number of samples can be observed in these two datasets, which indicates that the consensus graph learned by APLC has a well-defined block structure.

## Convergence and Time Comparison

As mentioned earlier, the objective value of ALPC decreases monotonically with alternating updates of the variables and the objective function is lower bounded. In this section, we

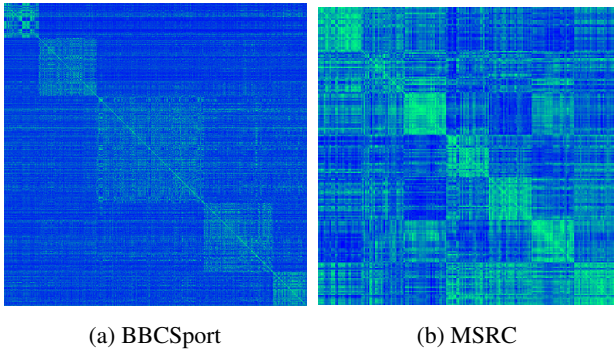


Figure 4: The visualization of the complete graph of ALPC on both datasets BBCSport and MSRC.

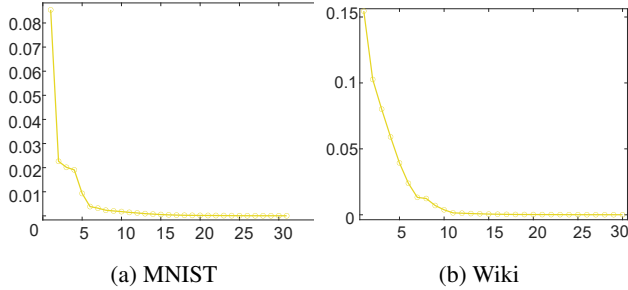


Figure 5: Convergence curves of ALPC on two datasets MNIST and Wiki.

experimentally demonstrate the convergence of the ALPC algorithm, and the convergence curves for MNIST and Wiki data are given in Figure 5. It is clear from the figure that the value of the objective function decreases rapidly and stabilizes at an iteration number of about 20, thus the method is highly convergent.

To be fair, we compared the computation time of ALPC and other methods on all datasets. The results are shown in Figure 6, where the y-axis is scaled by logarithms to mitigate the gap between the methods. It is clear from the figure 6 that ALPC is effective and takes clearly less computation time than the anchor-based MVC method, which proves that ALPC does not increase the computation time excessively.

### Ablation Study

While existing methods simply utilize the orthogonal constraint  $\mathbf{A}^{(v)T}\mathbf{A}^{(v)} = \mathbf{I}$  for dynamic learning of anchors, ALPC takes into account the real-world significance of anchors. In this section, we conduct ablation experiments to evaluate the effectiveness of the proposed shared latent clustering structure. We utilize Eq. 2 and impose regularization constraints on the anchor graph, which has been widely used in previous methods. We compared this variant (Ours-a) with our proposed ALPC method on all datasets and the results are shown in Table 4.

From the table, it can be clearly observed that the clustering performance of ALPC is better than that of Ours-a in all the datasets, which suggests that sharing the under-

Dataset	Method	ACC	NMI	Purity	F-score
MSRC	Ours-a	0.7507	0.7334	0.7688	0.7683
	Ours	<b>0.8819</b>	<b>0.7758</b>	<b>0.8819</b>	<b>0.7637</b>
BBCSport	Ours-a	0.8956	<b>0.8926</b>	0.8987	0.8614
	Ours	<b>0.9512</b>	0.8410	<b>0.9512</b>	<b>0.9009</b>
Wiki	Ours-a	0.5201	0.5254	0.5313	<b>0.5342</b>
	Ours	<b>0.6040</b>	<b>0.5514</b>	<b>0.6399</b>	0.5105
Caltech101-all	Ours-a	0.1007	0.1034	0.0889	0.0983
	Ours	<b>0.3284</b>	<b>0.3697</b>	<b>0.3617</b>	<b>0.2431</b>
MNIST	Ours-a	0.9108	0.9015	0.9026	0.9197
	Ours	<b>1.0000</b>	<b>0.9879</b>	<b>1.0000</b>	<b>0.9981</b>
YouTubeFace_sel	Ours-a	0.1530	0.0734	0.1688	0.0683
	Ours	<b>0.3046</b>	<b>0.2536</b>	<b>0.3299</b>	<b>0.1532</b>

Table 4: Comparison of the performance of Ours-a and Ours on six benchmark datasets.

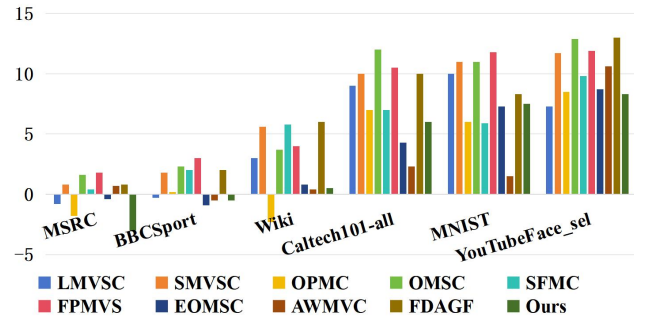


Figure 6: The relative running time of the compared algorithms and ALPC on the six benchmark datasets.

lying clustering structure helps in the generation of high-quality anchors, and the generated anchors share a consistent clustering structure with the original data, which results in the anchors learned by ALPC being more representative and discriminative. These reasons result in ALPC being able to learn higher quality anchor graphs, which leads to better clustering performance. From these ablation results, it is shown that sharing the underlying clustering structure is significant for the improvement of clustering performance.

### Conclusion

In this paper, we propose a new Anchor Learning with Potential Cluster Constraints for Multi-view Clustering method (ALPC). ALPC integrates and optimizes anchor selection and anchor graph learning into a unified framework. Unlike previous approaches, ALPC imposes shared potential clustering semantic constraints along with orthogonal constraints on anchors. Aligning the clustering centers of data clusters with the clustering centers of anchors constrains anchors to be generated uniformly in data clusters, making anchors more representative as well as discriminative. This allows for exploring inter-cluster diversity and intra-cluster consistency. ALPC truly realizes the relevance of anchors, and experiments demonstrate the effectiveness of the method.

## Acknowledgments

This work was supported by National Natural Science Foundation of China under Grant U21A20470, Grant 62172136 and Grant 72188101 and in part by the Institute of Advanced Medicine and Frontier Technology under Grant 2023 IHM01080, by National Key Research and Development Program of China Grant 2024YFB4710800, Liaoning Provincial Natural Science Foundation Grant 2024-MS-012.

## References

- Cao, X.; Zhang, C.; Fu, H.; Liu, S.; and Zhang, H. 2015. Diversity-induced Multi-view Subspace Clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 586–594.
- Chen, M.-S.; Huang, L.; Wang, C.-D.; Huang, D.; and Lai, J.-H. 2021. Relaxed multi-view clustering in latent embedding space. *Information Fusion*, 68: 8–21.
- Chen, M.-S.; Wang, C.-D.; Huang, D.; Lai, J.-H.; and Yu, P. S. 2022. Efficient Orthogonal Multi-view Subspace Clustering. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD '22*, 127–135. New York, NY, USA: Association for Computing Machinery. ISBN 9781450393850.
- Cui, C.; Ren, Y.; Pu, J.; Li, J.; Pu, X.; Wu, T.; Shi, Y.; and He, L. 2024. A novel approach for effective multi-view clustering with information-theoretic perspective. In *Proceedings of the 37th International Conference on Neural Information Processing Systems, NIPS '23*. Red Hook, NY, USA: Curran Associates Inc.
- Cui, C.; Ren, Y.; Pu, J.; Pu, X.; and He, L. 2023. Deep Multi-view Subspace Clustering with Anchor Graph. In Elkind, E., ed., *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, 3577–3585. International Joint Conferences on Artificial Intelligence Organization. Main Track.
- Huang, Z.; Ren, Y.; Pu, X.; and He, L. 2021. Non-Linear Fusion for Self-Paced Multi-View Clustering. In *Proceedings of the 29th ACM International Conference on Multimedia, MM '21*, 3211–3219. New York, NY, USA: Association for Computing Machinery. ISBN 9781450386517.
- Huang, Z.; Ren, Y.; Pu, X.; Huang, S.; Xu, Z.; and He, L. 2023. Self-supervised graph attention networks for deep weighted multi-view clustering. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence, AAAI'23/IAAI'23/EAAI'23*. AAAI Press. ISBN 978-1-57735-880-0.
- Jiang, G.; Peng, J.; Wang, H.; Mi, Z.; and Fu, X. 2022. 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.
- Kang, Z.; Zhou, W.; Zhao, Z.; Shao, J.; Han, M.; and Xu, Z. 2019. Large-scale Multi-view Subspace Clustering in Linear Time. *ArXiv*, abs/1911.09290.
- Li, X.; Zhang, H.; Wang, R.; and Nie, F. 2022. Multiview Clustering: A Scalable and Parameter-Free Bipartite Graph Fusion Method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1): 330–344.
- Ling, Y.; Chen, J.; Ren, Y.; Pu, X.; Xu, J.; Zhu, X.; and He, L. 2023. Dual label-guided graph refinement for multi-view graph clustering. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence, AAAI'23/IAAI'23/EAAI'23*. AAAI Press. ISBN 978-1-57735-880-0.
- Liu, J.; Liu, X.; Yang, Y.; Liu, L.; Wang, S.; Liang, W.; and Shi, J. 2021. One-pass Multi-view Clustering for Large-scale Data. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 12324–12333.
- Liu, S.; Liao, Q.; Wang, S.; Liu, X.; and Zhu, E. 2024. Robust and Consistent Anchor Graph Learning for Multi-View Clustering. *IEEE Transactions on Knowledge and Data Engineering*, 36(8): 4207–4219.
- Liu, S.; Wang, S.; Zhang, P.; Xu, K.; Liu, X.; Zhang, C.; and Gao, F. 2022. Efficient One-Pass Multi-View Subspace Clustering with Consensus Anchors. In *AAAI Conference on Artificial Intelligence*.
- Luo, S.; Zhang, C.; Zhang, W.; and Cao, X. 2018. Consistent and specific multi-view subspace clustering. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence, AAAI'18/IAAI'18/EAAI'18*. AAAI Press. ISBN 978-1-57735-800-8.
- Lv, W.; Zhang, C.; Li, H.; Jia, X.; and Chen, C. 2023. Joint Projection Learning and Tensor Decomposition-Based Incomplete Multiview Clustering. *IEEE Transactions on Neural Networks and Learning Systems*, 1–12.
- Qian, B.; Wang, Y.; Hong, R.; and Wang, M. 2023. Adaptive Data-Free Quantization. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 7960–7968.
- Ren, Y.; Kamath, U.; Domeniconi, C.; and Zhang, G. 2014. Boosted Mean Shift Clustering. In Calders, T.; Esposito, F.; Hüllermeier, E.; and Meo, R., eds., *Machine Learning and Knowledge Discovery in Databases*, 646–661. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN 978-3-662-44851-9.
- Sun, M.; Zhang, P.; Wang, S.; Zhou, S.; Tu, W.; Liu, X.; Zhu, E.; and Wang, C. 2021a. Scalable Multi-view Subspace Clustering with Unified Anchors. *Proceedings of the 29th ACM International Conference on Multimedia*.
- Sun, M.; Zhang, P.; Wang, S.; Zhou, S.; Tu, W.; Liu, X.; Zhu, E.; and Wang, C. 2021b. Scalable Multi-view Subspace Clustering with Unified Anchors. In *Proceedings of the 29th ACM International Conference on Multimedia, MM '21*, 3528–3536. New York, NY, USA: Association for Computing Machinery. ISBN 9781450386517.

Wan, X.; Liu, X.; Liu, J.; Wang, S.; Wen, Y.; Liang, W.; Zhu, E.; Liu, Z.; and Zhou, L. 2023. Auto-weighted Multi-view Clustering for Large-scale Data. In *AAAI Conference on Artificial Intelligence*.

Wang, H.; Yao, M.; Chen, Y.; Xu, Y.; Liu, H.; Jia, W.; Fu, X.; and Wang, Y. 2024a. Manifold-Based Incomplete Multi-View Clustering via Bi-Consistency Guidance. *IEEE Transactions on Multimedia*, 26: 10001–10014.

Wang, S.; Liu, X.; Zhu, X.; Zhang, P.; Zhang, Y.; Gao, F.; and Zhu, E. 2022a. Fast Parameter-Free Multi-View Subspace Clustering With Consensus Anchor Guidance. *IEEE Transactions on Image Processing*, 31: 556–568.

Wang, Y. 2021. Survey on Deep Multi-modal Data Analytics: Collaboration, Rivalry, and Fusion. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*.

Wang, Y.; Peng, J.; Wang, H.; and Wang, M. 2022b. Progressive learning with multi-scale attention network for cross-domain vehicle re-identification. *SCIENCE CHINA Information Sciences*, 65(6): 160103.

Wang, Y.; Qian, B.; Liu, H.; Rui, Y.; and Wang, M. 2024b. Unpacking the Gap Box Against Data-Free Knowledge Distillation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(9): 6280–6291.

Wang, Y.; Zhang, W.; Wu, L.; Lin, X.; Fang, M.; and Pan, S. 2016. Iterative views agreement: an iterative low-rank based structured optimization method to multi-view spectral clustering. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI'16*, 2153–2159. AAAI Press. ISBN 9781577357704.

Wei, X.-S.; Song, Y.-Z.; Aodha, O. M.; Wu, J.; Peng, Y.; Tang, J.; Yang, J.; and Belongie, S. J. 2021. Fine-Grained Image Analysis With Deep Learning: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44: 8927–8948.

Xia, W.; Gao, Q.; Wang, Q.; Gao, X.; Ding, C.; and Tao, D. 2023. Tensorized Bipartite Graph Learning for Multi-View Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4): 5187–5202.

Zhang, P.; Wang, S.; Li, L.; Zhang, C.; Liu, X.; Zhu, E.; Liu, Z.; Zhou, L.; and Luo, L. 2023. Let the Data Choose: Flexible and Diverse Anchor Graph Fusion for Scalable Multi-View Clustering. In *AAAI Conference on Artificial Intelligence*.