Learning Cluster-Wise Anchors for Multi-View Clustering

Chao Zhang[1], Xiuyi Jia[2], Zechao Li[2], Chunlin Chen[1], Huaxiong Li[1*]

[1]Department of Control Science and Intelligence Engineering, Nanjing University
[2]School of Computer Science and Engineering, Nanjing University of Science and Technology
chzhang@smail.nju.edu.cn, {jiaxy, zechao.li}@njust.edu.cn, {clchen, huaxiongli}@nju.edu.cn

Abstract

Due to its effectiveness and efficiency, anchor based multi-view clustering (MVC) has recently attracted much attention. Most existing approaches try to adaptively learn anchors to construct an anchor graph for clustering. However, they generally focus on improving the diversity among anchors by using orthogonal constraint and ignore the underlying semantic relations, which may make the anchors not representative and discriminative enough. To address this problem, we propose an adaptive Cluster-wise Anchor learning based MVC method, CAMVC for short. We first make an anchor cluster assumption that supposes the prior cluster structure of target anchors by pre-defining a consensus cluster indicator matrix. Based on the prior knowledge, an explicit cluster structure of latent anchors is enforced by learning diverse cluster centroids, which can explore both between-cluster diversity and within-cluster consistency of anchors, and improve the subspace representation discrimination. Extensive results demonstrate the effectiveness of our proposed method compared with some state-of-the-art MVC approaches.

Introduction

Due to the emergence of multi-source data in real-world, multi-view learning has recently attracted much attention, which aims to explore the underlying consistent and complementary information across views to improve the learning performance (Li, Yang, and Zhang 2019; Wen et al. 2023c; Sun et al. 2023; Zhang et al. 2023a; Jiang et al. 2023). As a fundamental task, multi-view clustering (MVC) tries to divide multi-view data into groups by discovering the intrinsic cluster structures without label information (Fang et al. 2023; Zhang et al. 2023b,c; Wen et al. 2023a,b).

Generally speaking, most existing MVC approaches can be divided into subspace based methods (Gao et al. 2015; Zhang et al. 2017; Gao et al. 2020; Huang et al. 2022), matrix factorization based methods (Liu et al. 2021; Wan et al. 2023), multi-kernel based methods (Liu 2023; Wan et al. 2022; Li et al. 2023a), and deep learning based methods (Lin et al. 2021; Xu et al. 2022; Wen et al. 2020; Yan et al. 2023). The majority of current MVC methods are based on subspace learning. Motivated by sparse representation that a sample can be approximately represented by the samples from the same class, multi-view subspace clustering learns a consensus self-expression matrix (usually sparse or low-rank), and then adopts spectral clustering on the similarity graph constructed from the self-expression matrix (Zhang et al. 2017; Lv et al. 2023; Zhang et al. 2022). However, multi-view subspace clustering usually suffers from high time complexity, $O(n^2)$ and even $O(n^3)$, due to the construction of $n \times n$ similarity graph and spectral clustering steps, making it unscalable to large datasets.

To address this problem, anchor based multi-view clustering is proposed, which leverages m ($m \ll n$) anchors to construct an $n \times m$ graph for clustering, and the time complexity can be greatly reduced to $O(mn)$ (Li et al. 2023b). The clustering performance highly depends on the quality of anchors, and how to determine representative anchors receives much attention. Some early methods randomly select a part of samples from original data as anchors, which is simple but usually unstable. Thus, some methods adopt heuristic strategies such as k-means (Li et al. 2015; Kang et al. 2020) and feature score based schemes (Li et al. 2022; Xia et al. 2023). For example, (Li et al. 2022) proposed a directly alternate sampling scheme for anchor selection by measuring feature score. Instead of directly selecting anchors from original samples as mentioned above, adaptive anchor learning has become popular recently. It usually adaptively learns latent anchors and subspace representation in a unified model. For example, (Sun et al. 2021) adaptively learns a consensus anchor matrix and a non-negative graph for all views in a latent subspace. (Chen et al. 2022) incorporates k-means into adaptive anchor learning to directly learn a cluster indicator for anchor graph.

Although adaptive anchor learning based MVC methods have achieved promising performance, they generally impose orthogonal constraint on anchors to improve the anchor diversity, and jointly construct an anchor graph to reconstruct the original samples. However, the orthogonal constraint may cause an imbalanced anchor distribution, i.e., there may be few and even no anchors for some data clusters. To obtain a discriminative representation matrix, it is expected that the anchors consist of some representative samples from each cluster and the samples can be only reconstructed by those anchors that come from the same class. In other words, the learned anchors have a balanced and dis-

*Corresponding author.

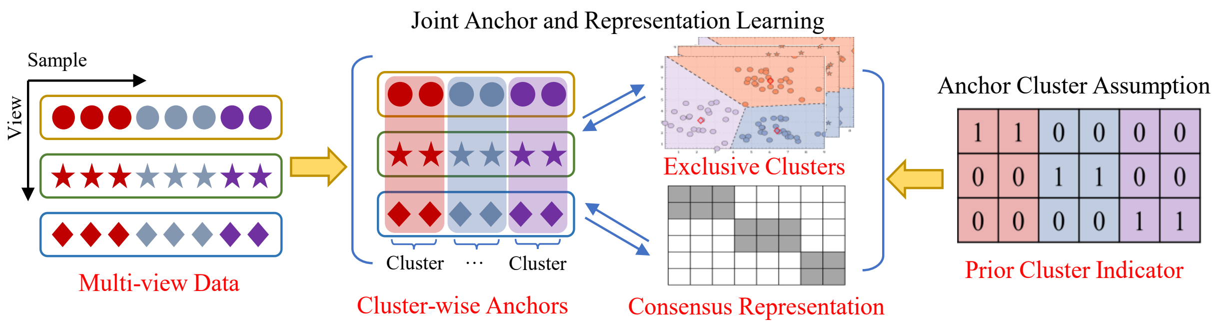

Figure 1: Framework of the proposed CAMVC method. CAMVC integrates latent anchors learning and consensus subspace representation construction into a unified model. Under the anchor cluster assumption, a prior cluster indicator matrix is pre-defined to guide anchor learning. The latent anchors are enforced to have an exclusive cluster structure with within-cluster consistency and between-cluster diversity.

criminative distribution that shows a natural cluster structure matching the original data. The within-cluster anchors are semantically correlated and between-cluster anchors are semantically uncorrelated, such that these representative anchors can describe the multi-class data distributions and the learned subspace representation can be more discriminative.

Motivated by this, we propose a Cluster-wise Anchor learning method for MVC (CAMVC), which enforces an explicit cluster structure of latent anchors for enhancing the subspace representation discrimination. Figure 1 illustrates the framework of our method. In specific, CAMVC adaptively learns multi-view anchors and a consensus subspace representation in a unified optimization model. With anchor cluster assumption, a prior cluster indicator matrix is pre-defined, which can be viewed as a label matrix, and it is used to guide the adaptive anchor learning. An explicit anchor cluster structure is enforced to match the original data. Matrix factorization is applied to learn orthogonal centroids for anchor clusters, such that within-cluster anchors are semantically correlated and between-cluster anchors exclusive, which benefits the discriminative subspace representation learning. The main contributions of this paper are summarized as follows:

- A novel anchor based multi-view clustering method is proposed, which adaptively learns discriminative anchors with cluster structure regularization and discriminative representation in a unified optimization framework.
- An anchor cluster assumption is introduced by defining a prior cluster indicator. Diverse centroids of anchor clusters are learned to ensure the within-cluster semantic consistency and between-cluster diversity.
- An alternating optimization algorithm is designed to solve the proposed model. Extensive experiments demonstrate the effectiveness of our method compared with state-of-the-art approaches.

Related Work

In this section, we introduce the multi-view subspace clustering and anchor based MVC methods. Table 1 lists the main used notations and descriptions.

Notation	Description
n, v, k	Number of samples, views, clusters
m	Number of anchors in each cluster
d_i	Feature dimension of the i-th view
$\mathbf{X}^i \in \mathbb{R}^{d_i \times n}$	Data matrix of the i-th view
$\mathbf{A}^i \in \mathbb{R}^{d_i \times mk}$	Anchor matrix of the i-th view
$\mathbf{Z} \in \mathbb{R}^{mk \times n}$	Consensus subspace representation
$\mathbf{H}^i \in \mathbb{R}^{d_i \times k}$	Anchor cluster centroids of view i
$\mathbf{Y} \in \{0,1\}^{k \times mk}$	Prior anchor cluster indicator

Table 1: Main notations and descriptions.

Multi-view Subspace Clustering

Multi-view subspace clustering (MVSC) methods generally adopt self-expression to learn an affinity matrix, and most of them can be described by the following framework:

$$\min_{\mathbf{Z}^i,\mathbf{Z}} \sum_{i=1}^{v} \|\mathbf{X}^i - \mathbf{X}^i\mathbf{Z}^i\|_F^2 + \lambda\Omega(\mathbf{Z}, \mathbf{Z}^i), \tag{1}$$

where $\mathbf{Z}^i$ denotes subspace representation of the i-th view, and $\Omega(\cdot)$ represents the unified regularization term to obtain a consensus representation $\mathbf{Z}$.

Many existing MVSC methods are developed from the above model. For example, (Gao et al. 2015) proposes to learn a subspace representation for each view, and a common cluster structure is simultaneously obtained based on spectral clustering framework. (Luo et al. 2018) decomposes the subspace representation of each view into an invariant part and an exclusive part to explore the consistent and complementary information across views. Moreover, to enhance the subspace representation discrimination, various regularization terms are imposed on subspace representation, typically including low-rank constraints (Zhang et al. 2017; Brbic and Kopriva 2018) and sparse constraints (Lu, Yan, and Lin 2016; Elhamifar and Vidal 2013). Recently, tensor low-rank based regularization becomes popular, which regards the subspace representation matrices of different views as a tensor to explore the high-order low-rank property, such as (Gao et al. 2020; Xie et al. 2018; Wu, Lin, and Zha 2019).

However, these methods construct a full $n \times n$ similarity graph and perform spectral clustering with $O(n^3)$ complexity, making it unapplicable to large-scale datasets.

Anchor Based Multi-view Clustering

Anchor based multi-view clustering received much attention to handle large-scale datasets (Li et al. 2015; Kang et al. 2020; Wang et al. 2022; Li et al. 2022; Qiang et al. 2021; Yang et al. 2023; Kang et al. 2022). The principle of anchor based method is to construct an $n \times m$ similarity graph between n samples and m anchors to replace full $n \times n$ graph, which greatly reduces the time complexity and maintains considerable clustering performance.

Most existing anchor based MVC methods can be divided into anchor sampling based approaches and anchor learning based approaches. The former first adopts heuristic strategies to select anchors from original samples and then constructs anchor graphs, while the latter adaptively learns anchors and graphs in a unified framework. (Kang et al. 2020) first adopts k-means to select anchors, and then constructs an anchor graph for each view. (Wang et al. 2022) proposes a parameter-free method by simultaneously learning a consensus anchor matrix and an anchor graph in a low-dimensional subspace. (Liu et al. 2022) jointly learns consensus latent anchors and a low-rank graph with exactly k connected components.

Although these methods achieved considerable performance, there are still some limitations. Anchor sampling based approaches separate the anchor selection and graph construction into two steps, and the learned representation may be sub-optimal due to the lack of mutual negotiation. Anchor learning based approaches unify the two steps, but they only focus on the anchor diversity and neglect the underlying semantic relations. Therefore, this paper proposes an adaptive cluster-wise anchor learning based method by exploring both semantic anchor consistency and diversity.

The Proposed Method

In this section, we will introduce our proposed CAMVC method in detail, including model formulation, optimization, algorithm complexity and convergence analysis.

Formulation

Instead of performing heuristic sampling, we intend to adaptively learn anchors and graphs by optimization. Based on the assumption that heterogeneous data in different views share a consensus low-dimensional subspace (Zhang et al. 2017), we solve the following problem to obtain view-specific anchors and a consensus subspace representation,

$$\min_{\mathbf{A}^i,\mathbf{Z}} \sum_{i=1}^{v} \|\mathbf{X}^i - \mathbf{A}^i\mathbf{Z}\|_F^2 + \beta\|\mathbf{Z}\|_F^2. \tag{2}$$

Learning representative anchors is the key to obtain discriminative representation and improve clustering performance. Previous methods generally adopt orthogonal constraint, i.e., ${\mathbf{A}^i}^T\mathbf{A}^i = \mathbf{I}$, to improve anchor diversity. However, the learned pairwise exclusive anchors may be not representative to well reconstruct the complex data $\mathbf{X}^i$. To obtain a discriminative representation, it is reasonable to expect that each sample $\mathbf{x}^i$ can be well reconstructed by a few anchors $\{\mathbf{a}_1^i, \mathbf{a}_2^i, ..., \mathbf{a}_m^i\}$ that are semantically correlated with $\mathbf{x}^i$, and thus these anchors show a clear cluster structure that matches the original samples, as shown in Figure 2, where the cluster-wise anchors are promising to learn more discriminative subspace representation for clustering. Inspired by this, we make the following *anchor cluster assumption* to seek cluster-wise anchors.

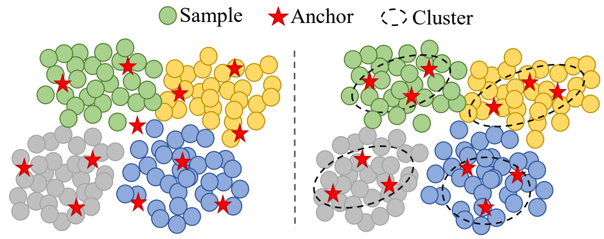

Figure 2: Non-cluster-wise (left) anchors and cluster-wise (right) anchors.

Assumption 1 (Anchor Cluster Assumption). *Given multi-view data* $\mathbf{X} = \{\mathbf{X}^i\}_{i=1}^v$*, it is assumed that the latent multi-view anchors* $\mathbf{A} = \{\mathbf{A}^i\}_{i=1}^v$ *have a consensus cluster structure* $\{C_1, C_2, ..., C_k\}$ *that matches the cluster structure of original data* $\mathbf{X}$.

Based on Assumption 1, if we want to divide $\mathbf{X}$ into k clusters, it is expected that the anchors $\mathbf{A}$ can also be divided into the same k clusters. Without loss of generality, we assume each anchor cluster C_i contains m representative anchors $\{\mathbf{a}_1^i, \mathbf{a}_2^i, ..., \mathbf{a}_m^i\}$ $(i = 1, ..., v)$, and the total number of anchors is mk. We define a prior cluster indicator matrix $\mathbf{Y} \in \{0,1\}^{k \times mk}$, where $y_{ij} = 1$ if the j-th anchor belongs to the i-th cluster and 0 otherwise. Then, we solve the following problem to enforce the cluster structure of $\{\mathbf{A}^i\}_{i=1}^v$:

$$\min_{\mathbf{A}^i,\mathbf{H}^i} \sum_{i=1}^{v} \|\mathbf{A}^i - \mathbf{H}^i\mathbf{Y}\|_F^2 \ \ s.t.\ {\mathbf{H}^i}^T\mathbf{H}^i = \mathbf{I}. \tag{3}$$

Eq. (3) is equivalent to dividing the anchors into k predefined clusters, $\mathbf{H}^i \in \mathbb{R}^{d_i \times k}$ contains the k anchor cluster centroids, and the orthogonal constraint ensures the between-cluster diversity. Eq. (3) enforces the within-cluster anchors semantically correlated and between-cluster anchors exclusive. Combining Eq. (2) and Eq. (3). we formulate the objective function of CAMVC as follows:

$$\min_{\Omega} \sum_{i=1}^{v} \left(\|\mathbf{X}^i - \mathbf{A}^i\mathbf{Z}\|_F^2 + \alpha\|\mathbf{A}^i - \mathbf{H}^i\mathbf{Y}\|_F^2\right) + \beta\|\mathbf{Z}\|_F^2$$
$$s.t.\ {\mathbf{H}^i}^T\mathbf{H}^i = \mathbf{I}. \tag{4}$$

where α and β are balance parameters. Eq. (4) combines latent cluster-wise anchors and discriminative representation learning into a unified framework. The first term aims to learn representative anchors to reconstruct the samples, the second term enforces the anchor cluster structure by prior

Algorithm 1: **CAMVC algorithm**

Input: Multi-view data $\{\mathbf{X}^i\}_{i=1}^v$, cluster number k, parameters α, β, m.

1: Initialize $\mathbf{A}^i = \mathbf{0}, \mathbf{H}^i = \mathbf{0}$.
2: Construct the prior cluster indicator matrix $\mathbf{Y}$.
3: **while** not converged **do**
4: Update $\mathbf{Z}$ by Eq. (7);
5: Update $\{\mathbf{A}^i\}_{i=1}^v$ by Eq. (9);
6: Update $\{\mathbf{H}^i\}_{i=1}^v$ by solving (11);
7: **end while**

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

cluster indicator, and the last term is to avoid overfitting. An orthogonal constraint is imposed on the anchor cluster centers to improve the anchor semantic exclusivity among different clusters.

Optimization

To solve the CAMVC model, we adopt an alternating optimization strategy. In each iteration, all variables are updated alternately with other fixed.

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

$$\min_{\mathbf{Z}} \sum_{i=1}^{v} \|\mathbf{X}^i - \mathbf{A}^i\mathbf{Z}\|_F^2 + \beta\|\mathbf{Z}\|_F^2. \tag{5}$$

The above problem can be solved by setting its derivative over $\mathbf{Z}$ to zero, i.e.,

$$-2\sum_{i=1}^{v} \mathbf{A}^{i^T}\mathbf{X}^i + 2\sum_{i=1}^{v} \mathbf{A}^{i^T}\mathbf{A}^i\mathbf{Z} + 2\beta\mathbf{Z} = 0. \tag{6}$$

Obviously, the optimal solution is

$$\mathbf{Z} = \left(\sum_{i=1}^{v} \mathbf{A}^{i^T}\mathbf{A}^i + \beta\mathbf{I}\right)^{-1} \left(\sum_{i=1}^{v} \mathbf{A}^{i^T}\mathbf{X}^i\right). \tag{7}$$

Update $\mathbf{A}^i$ Because each $\mathbf{A}^i$ $(i = 1, ..., v)$ is independent from each other in terms of different views, and we can solve each $\mathbf{A}^i$ independently. With fixing other variables, the sub-problem w.r.t. $\mathbf{A}^i$ is

$$\min_{\mathbf{A}^i} \|\mathbf{X}^i - \mathbf{A}^i\mathbf{Z}\|_F^2 + \alpha\|\mathbf{A}^i - \mathbf{H}^i\mathbf{Y}\|_F^2. \tag{8}$$

Similar to $\mathbf{Z}$, by setting the derivative over $\mathbf{A}^i$ to zero, we can obtain the optimal solution as follows:

$$\mathbf{A}^i = \left(\mathbf{X}^i\mathbf{Z}^T + \alpha\mathbf{H}^i\mathbf{Y}\right)\left(\mathbf{Z}\mathbf{Z}^T + \alpha\mathbf{I}\right)^{-1}. \tag{9}$$

Update $\mathbf{H}^i$ With other variables fixed, the optimization problem w.r.t. $\mathbf{H}^i$ is

$$\min_{\mathbf{H}^i} \|\mathbf{A}^i - \mathbf{H}^i\mathbf{Y}\|_F^2 \;\; s.t. \; \mathbf{H}^{i^T}\mathbf{H}^i = \mathbf{I}. \tag{10}$$

It is equivalent to solving

$$\max_{\mathbf{H}^i} \mathrm{Tr}\left(\mathbf{H}^{i^T}\mathbf{A}^i\mathbf{Y}^T\right) \;\; s.t. \; \mathbf{H}^{i^T}\mathbf{H}^i = \mathbf{I}. \tag{11}$$

Dataset	n	v	k	d
MSRC	210	4	7	24/512/256/254
BBCSport	544	2	5	3183/3203
Wiki	2866	2	10	128/10
Notting-Hill	4660	3	5	6750/2000/3304
Caltech101	8677	4	101	2048/4800/3540/1240
Fashion	10000	3	10	784/784/784
MNIST	60000	3	10	342/1024/64

Table 2: General statistics of datasets, where n, v, k, d denote the number of samples, views, classes, and feature dimension, respectively.

It is an orthogonal Procrustes problem (Wen and Yin 2013), and can be solved by the Singular Value Decomposition (SVD) of $\mathbf{A}^i\mathbf{Y}^T$, i.e., $\mathbf{A}^i\mathbf{Y}^T = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$. The optimal solution is $\mathbf{H}^i = \mathbf{U}\mathbf{V}^T$.

We can see that CAMVC only needs to update three variables and each sub-problem can be solved by an efficient solution. For convenience, the whole optimization process is summarized in Algorithm 1.

Complexity and Convergence Analysis

In this section, we analyze the computational complexity including space complexity and time complexity, and convergence property of our CAMVC.

Complexity *1) Time Complexity:* The time consumption mainly consists of the update of each variable. For updating $\mathbf{Z}$, the time cost depends on matrix inverse and multiplication, and the time complexity is $O(m^3k^3 + m^2k^2d + m^2k^2n + mkdn)$ where $d = \sum_{i=1}^{v} d_i$. For updating $\mathbf{A}^i$, the time complexity is $O(m^3k^3 + d_imkn)$. For updating $\mathbf{H}^i$, it takes $O(d_imk^2)$ complexity. Th post-processing of k-means also has linear complexity w.r.t. n. Since $m, k \ll n$, the overal time complexity of CAMVC is linear to the number of samples, $O(n)$, which enables it to handle large-scale datasets.

2) Space Complexity: The major space cost of CAMVC are matrices $\mathbf{Z} \in \mathbb{R}^{mk\times n}$, $\mathbf{A}^i \in \mathbb{R}^{d_i\times n}$, $\mathbf{H}^i \in \mathbb{R}^{d_i\times k}$ and matrix multiplication results. The space complexity of CAMVC is $O(m^2k^2 + dmk + (mk + d)n)$, i.e., linear to n.

Convergence The whole objective function is not convex w.r.t. all variables. We adopt an alternating optimization algorithm to update each variable with an analytical solution. Let $\mathcal{J}(\mathbf{Z}_t, \mathbf{A}_t, \mathbf{H}_t)$ denote the objective function value at the t-th iteration, it can be obtained that $\mathcal{J}(\mathbf{Z}_t, \mathbf{A}_t, \mathbf{H}_t) \leq \mathcal{J}(\mathbf{Z}_{t+1}, \mathbf{A}_t, \mathbf{H}_t) \leq \mathcal{J}(\mathbf{Z}_{t+1}, \mathbf{A}_{t+1}, \mathbf{H}_t) \leq \mathcal{J}(\mathbf{Z}_{t+1}, \mathbf{A}_{t+1}, \mathbf{H}_{t+1})$, indicating the monotonically decrease of objective function value. Since it is lower bounded by zero, the algorithm can be verified to converge to a local minimum.

Experiment

In this section, we conduct experiments to compare our CAMVC with some state-of-the-art methods, and analyzes the algorithm performance.

Dataset	LMVSC	SMVSC	OPMC	OMSC	SFMC	FPMVS	EOMSC	AWMVC	FDAGF	Ours
					ACC					
MSRC	0.3476	0.6952	0.6286	0.7048	0.7286	0.6047	0.5905	0.7810	0.6429	**0.8418**
BBCSport	0.6544	0.5091	0.6691	0.4522	0.8254	0.4209	0.4154	0.6397	0.4669	**0.8997**
Wiki	0.1884	0.3273	0.1766	0.3535	0.3039	0.3140	**0.5415**	0.2170	0.5265	0.5386
Notting-Hill	0.7182	0.8058	0.6751	0.7142	0.7433	0.7835	0.8513	0.9255	0.5382	**0.9521**
Caltech101	0.5364	0.4681	0.4387	0.4237	0.2691	0.4797	0.4818	0.5676	0.6025	**0.6301**
Fashion	0.6160	0.6952	0.7894	0.6643	0.7818	0.6397	0.6022	0.7305	0.6454	**0.7897**
MNIST	0.9896	0.9884	0.8392	0.8922	**0.9905**	0.9884	0.9808	0.9885	0.9887	0.9826
Ave. Score	0.5787	0.6413	0.6024	0.6007	0.6632	0.6044	0.6376	0.6928	0.6302	**0.8049**
Ave. Rank	6.42	5.28	7.00	6.85	4.57	6.85	6.57	4.00	5.42	**2.00**
					NMI					
MSRC	0.2461	0.6190	0.5533	0.6314	0.6987	0.5557	0.4325	0.7160	0.6207	**0.8370**
BBCSport	0.4471	0.2118	0.6705	0.2129	0.7770	0.1508	0.2367	0.4820	0.3749	**0.8951**
Wiki	0.0469	0.1628	0.0454	0.1991	0.3196	0.1715	0.5290	0.0783	0.5149	**0.5339**
Notting-Hill	0.6890	0.7319	0.7126	0.6587	0.8544	0.6748	0.7247	0.8387	0.4511	**0.9624**
Caltech101	0.8007	0.6171	0.6683	0.5935	0.4273	0.6323	0.6464	0.8067	0.8182	**0.8426**
Fashion	0.5504	0.7554	0.8213	0.7713	**0.8642**	0.7340	0.7207	0.7212	0.7127	0.7817
MNIST	0.9685	0.9650	0.9134	0.9336	0.9709	0.9650	0.9475	0.9647	0.9658	**0.9787**
Ave. Score	0.5355	0.5804	0.6264	0.5715	0.7017	0.5549	0.6054	0.6582	0.6369	**0.8331**
Ave. Rank	6.85	6.28	6.28	6.85	3.42	7.14	6.42	4.85	5.57	**1.28**

Table 3: Clustering performance of different methods w.r.t. *ACC* and *NMI* on all datasets. The best results are highlighted in bold, and the second-best results are underlined.

Experimental Setup

We conduct experiments on seven popular datasets, including MSRC (Chen et al. 2021), BBCSport[1], Wiki[2], Notting-Hill (Chen et al. 2021), Caltech101[3], Fashion (Xiao, Rasul, and Vollgraf 2017), and MNIST[4]. Table 2 shows the general statistics of these datasets.

We adopt nine state-of-the-art MVC methods as baselines, including **LMVSC** (Kang et al. 2020), **SMVSC** (Sun et al. 2021), **OPMC** (Liu et al. 2021), **OMSC** (Chen et al. 2022), **SFMC** (Li et al. 2022), **FPMVS** (Wang et al. 2022), **EOMSC** (Liu et al. 2022), **AWMVC** (Wan et al. 2023), and **FDAGF** (Zhang et al. 2023d), in which OPMC and AWMVC are based on matrix factorization, and the others are anchor based MVC methods.

For a fair comparison, we use the official codes of baselines. Since CAMVC and most baselines need k-means to obtain final clusters, following (Wang et al. 2022; Wan et al. 2023), we run 50 times k-means on final representation to report the best results. The optimal parameters of baselines are tuned by grid search in suggested ranges. For our CAMVC, we tune α in the range of $\{10^{-3}, 10^{-2}, ..., 10^{1}\}$, β in $\{10^{-1}, 10^{-2}, ..., 10^{3}\}$, and m in $\{1, 3, 5\}$. We adopt four popular evaluation metrics including accuracy (ACC), normalized mutual information (NMI), purity, and Fscore. For all metrics, the higher value indicates the better performance. All experiments are conducted using MATLAB 2017b with i5-1230 CPU and 16GB RAM.

[1] http://mlg.ucd.ie/datasets/bbc.html

[2] http://www.svcl.ucsd.edu/projects/crossmodal/

[3] https://paperswithcode.com/dataset/caltech-101

[4] http://yann.lecun.com/exdb/mnist/

Clustering Results

We compare our method with baselines on seven datasets w.r.t. four metrics. Table 3 reports the clustering results over ACC and NMI, and Table 4 reports the results over Purity and Fscore. The average scores and ranks on all datasets are also listed. From Table 3 and Table 4, we can obtain the following observations:

- Our CAMVC achieves the best and second-best results in most cases, and it obtains the highest average scores and ranks w.r.t. all metrics. For ACC metric, the average rank of CAMVC is 2.00, while the second-best rank is only 4.00. These results demonstrate the effectiveness and superiority of our method compared with the state-of-the-art approaches.
- Compared with LMVSC, some methods like SMVSC, FPMVS, EOMSC and our CAMVC can obtain better performance. The main reason is that the former adopts the static anchor strategy, and the latter dynamically learns anchors to obtain an affinity matrix for clustering, which is more flexible to construct a discriminative similarity matrix for clustering.
- On some datasets like MSRC, BBCSport and Notting-Hill, our method significantly outperforms most related anchor based baselines, such as SMVSC, FPMVC, FDAGF. It is because our method considers the anchor cluster structure by introducing a prior cluster indicator

Dataset	LMVSC	SMVSC	OPMC	OMSC	SFMC	FPMVS	EOMSC	AWMVC	FDAGF	Ours
					Purity					
MSRC	0.4048	0.6952	0.6381	0.7381	0.7667	0.6190	0.6048	0.7810	0.7571	**0.8466**
BBCSport	0.6636	0.5367	0.7445	0.5110	0.8566	0.5183	0.5037	0.6893	**0.9228**	0.9070
Wiki	0.2083	0.3433	0.2069	0.3765	0.3458	0.3367	0.6141	0.2547	**0.6811**	0.5434
Notting-Hill	0.8101	0.8305	0.7942	0.7807	0.8723	0.7836	0.8513	0.9255	0.8790	**0.9420**
Caltech101	0.7206	0.5071	0.4897	0.4891	0.3575	0.5217	0.5345	0.7701	0.7489	**0.8183**
Fashion	0.6178	0.6988	0.8287	0.7070	0.7820	0.6723	0.6333	0.7305	0.7846	**0.8321**
MNIST	0.9896	0.9884	0.9385	0.8922	**0.9905**	0.9884	0.9808	0.9885	0.9868	0.9826
Ave. Score	0.6307	0.6571	0.6629	0.6421	0.7102	0.6342	0.6746	0.7342	0.8229	**0.8389**
Ave. Rank	6.85	6.14	6.85	7.57	4.28	7.28	6.85	3.85	3.00	**2.28**
					Fscore					
MSRC	0.2480	0.5766	0.5155	0.6123	0.6236	0.5029	0.4261	0.6852	0.5337	**0.8454**
BBCSport	0.4854	0.3645	0.6501	0.3556	0.7829	0.3274	0.3468	0.5234	0.4978	**0.8683**
Wiki	0.1294	0.2103	0.1244	0.2243	0.2113	0.2146	0.4832	0.1432	0.4325	**0.5547**
Notting-Hill	0.6980	0.7832	0.6788	0.6560	0.7670	0.7367	0.7585	0.8363	0.4873	**0.9350**
Caltech101	0.3946	**0.5286**	0.3863	0.3986	0.0667	0.5131	0.5278	0.4446	0.4725	0.4607
Fashion	0.4914	0.6738	0.7536	0.6659	0.7288	0.6272	0.6066	0.6573	0.6256	**0.8369**
MNIST	0.9794	0.9768	0.8550	0.8923	**0.9813**	0.9768	0.9621	0.9671	0.9735	0.9787
Ave. Score	0.4895	0.5877	0.5662	0.5436	0.5945	0.5570	0.5873	0.6082	0.5747	**0.7828**
Ave. Rank	7.42	4.42	7.00	6.57	4.14	6.28	6.28	5.00	6.00	**1.85**

Table 4: Clustering performance of different methods w.r.t. *Purity* and *Fscore* on all datasets.

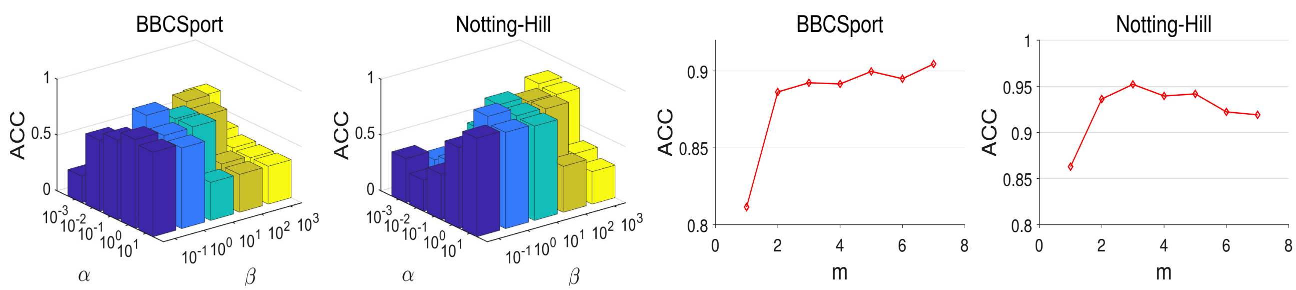

Figure 3: ACC values of CAMVC w.r.t. α and β on BBCSport and BDGP datasets.

to supervise anchor learning, and thus our method can obtain more discriminative similarity matrix.

Parameter Analysis

The CAMVC model contains two parameters α and β, which controls the anchor cluster penalty and subspace representation smoothness, respectively. In this subsection, we study the influence of them by grid search. In detail, we define two ranges $\{10^{-3}, 10^{-2}, ..., 10^{1}\}$, $\{10^{-1}, 10^{-2}, ..., 10^{3}\}$ for α and β, respectively, and record the clustering performance of CAMVC with different combinations of the two parameters. Figure 3 shows the ACC values of CAMVC on BBCSport and Notting-Hill datasets with different combinations of α and β. We can observe that when α and β are selected from $[10^{-1}, 10^{1}]$, CAMVC can obtain satisfactory performance. Besides, when α is too small, the results are not optimal, which indicates that the anchor cluster structure regularization term takes effects to improve the algorithm performance. To obtain cluster-wise anchors, our method assumes m anchors for each cluster. We also investigate the influence of m value and the results are shown in Figure 3. It can be observed that our method can obtain satisfactory performance when m is selected from $[2, 5]$. Besides, when $m = 1$, the clustering results are not desirable, which implies that one anchor for each cluster is not enough to well reconstruct the samples and the learned anchor graph is also not optimal.

Convergence and Time Comparison

As illustrated before, the objective value of CAMVC decreases monotonically with variables alternate update and the objective function is lower-bounded. In this subsection, we experimentally prove the convergence of CAMVC algorithm. Figure 4 plots the convergence curves on BBCSport and Notting-Hill datasets. It is seen that the objective function value decreases monotonously and the ACC value gradually increases w.r.t. iterations, demonstrating the good convergence property.

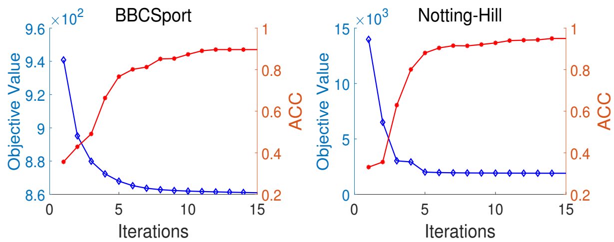

Figure 4: Convergence curves of CAMVC.

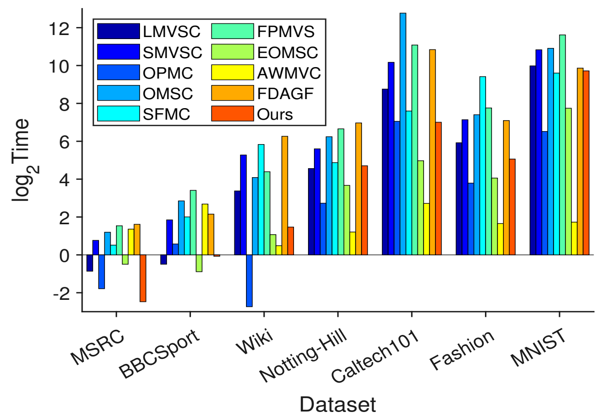

Figure 5: Running time comparison of different methods.

In addition, to investigate the efficiency of CAMVC, we compare the running time of CAMVC and baselines on all datasets. The results are shown in Figure 5, where the y axis is scaled by $\log$ to mitigate the gap between methods. Since all baselines and CAMVC have $O(n)$ time complexity, they are scalable on all datasets. Our method is efficient and surpasses some baselines such as OMSC, SMVSC, FPMVS, FDAGF. The main reason is that our method only needs to update three variables and each update has an efficient closed-form solution, while those baselines compute the anchor graph matrix column by column. Matrix factorization based approaches, OPMC and AWMVC, are more efficient than our CAMVC. It is because that they directly learn the common latent representation by matrix factorization. Although they are more efficient, CAMVC can achieve better clustering performance in all cases. Generally speaking, the execution speed of CAMVC is satisfactory and scalable to large-scale datasets.

Ablation Study

Instead of learning pairwise orthogonal anchors to construct an affinity matrix in previous methods, our CAMVC assumes the anchor cluster structure to seek cluster-wise latent anchors. In this subsection, we conduct ablation study to evaluate the effectiveness of cluster structure regularization. Based on CAMVC, we derive a variant by discarding the second term in Eq. (4), i.e., $\alpha = 0$, and adding the orthogonal constraint $\mathbf{A}^{i^T}\mathbf{A}^i = \mathbf{I}$, which is widely used in previous methods. We compare this variant (Ours-a) with our original CAMVC on all datasets, and the results are shown in Table 5. We can observe that Ours achieves better performance than Ours-a in all cases, especially on MSRC, Notting-Hill, Caltech101, and Fashion datasets, where Ours outperforms Ours-a with significant gaps. We also make a t-SNE visualization (Van der Maaten and Hinton 2008) comparison of learned representation $\mathbf{Z}$ on Notting-Hill. Ours reveals a relatively more discriminative cluster structure than Ours-a, resulting in better performance. These ablation results demonstrate that the anchor cluster assumption helps to learn more discriminative subspace representation.

Dataset	Method	ACC	NMI	Purity	Fscore
MSRC	Ours-a	0.7507	0.7334	0.7688	0.7683
	Ours	**0.8418**	**0.8370**	**0.8466**	**0.8454**
BBCSport	Ours-a	0.8956	0.8926	0.8987	0.8614
	Ours	**0.8997**	**0.8951**	**0.9070**	**0.8683**
Wiki	Ours-a	0.5201	0.5254	0.5313	0.5342
	Ours	**0.5386**	**0.5339**	**0.5434**	**0.5547**
Notting-Hill	Ours-a	0.9254	0.9336	0.9174	0.8877
	Ours	**0.9521**	**0.9624**	**0.9420**	**0.9350**
Caltech101	Ours-a	0.5597	0.7933	0.7647	0.4367
	Ours	**0.6301**	**0.8426**	**0.8183**	**0.4607**
Fashion	Ours-a	0.7384	0.7343	0.8281	0.8089
	Ours	**0.7897**	**0.7817**	**0.8321**	**0.8369**
MNIST	Ours-a	0.9808	0.9715	0.9726	0.9597
	Ours	**0.9826**	**0.9787**	**0.9826**	**0.9787**

Table 5: Ablation study results.

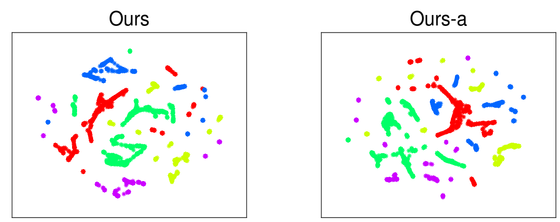

Figure 6: Representation visualization on Notting-Hill.

Conclusion

In this paper, we propose a novel Cluster-wise Anchor based Multi-view Clustering (CAMVC) method. CAMVC integrates latent anchor and discriminative subspace representation learning into a unified optimization model. Instead of learning orthogonal anchors, CAMVC seeks cluster-wise latent anchors for discriminative subspace representation. An anchor cluster assumption is introduced by pre-defining a consensus cluster indicator matrix. With the prior supervision, an explicit cluster structure of anchors is enforced by learning diverse cluster centroids, which can explore both between-cluster diversity and within-cluster consistency. Experiments demonstrate the effectiveness and efficiency of our proposed method.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China under Grants 62176116, 62073160, U21B2043, the National Key Research and Development Program of China under Grant 2022ZD0118802, and the Natural Science Foundation of the Jiangsu Higher Education Institutions of China under Grant 20KJA520006.

References

Brbic, M.; and Kopriva, I. 2018. Multi-view low-rank sparse subspace clustering. *Pattern Recognition*, 73: 247–258.

Chen, M.; Huang, L.; Wang, C.; Huang, D.; and Lai, J. 2021. Relaxed multi-view clustering in latent embedding space. *Inf. Fusion*, 68: 8–21.

Chen, M.; Wang, C.; Huang, D.; Lai, J.; and Yu, P. S. 2022. Efficient Orthogonal Multi-view Subspace Clustering. In *ACM SIGKDD*, 127–135.

Elhamifar, E.; and Vidal, R. 2013. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE TPAMI*, 35(11): 2765–2781.

Fang, U.; Li, M.; Li, J.; Gao, L.; Jia, T.; and Zhang, Y. 2023. A Comprehensive Survey on Multi-view Clustering. *IEEE TKDE*.

Gao, H.; Nie, F.; Li, X.; and Huang, H. 2015. Multi-view Subspace Clustering. In *ICCV*, 4238–4246.

Gao, Q.; Xia, W.; Wan, Z.; Xie, D.; and Zhang, P. 2020. Tensor-SVD Based Graph Learning for Multi-View Subspace Clustering. In *AAAI*, 3930–3937.

Huang, S.; Wu, H.; Ren, Y.; Tsang, I. W.; Xu, Z.; Feng, W.; and Lv, J. 2022. Multi-view Subspace Clustering on Topological Manifold. In *NeurIPS*.

Jiang, B.; Zhang, C.; Zhong, Y.; Liu, Y.; Zhang, Y.; Wu, X.; and Sheng, W. 2023. Adaptive collaborative fusion for multi-view semi-supervised classification. *Inf. Fusion*, 96: 37–50.

Kang, Z.; Lin, Z.; Zhu, X.; and Xu, W. 2022. Structured Graph Learning for Scalable Subspace Clustering: From Single View to Multiview. *IEEE TCYB*, 52(9): 8976–8986.

Kang, Z.; Zhou, W.; Zhao, Z.; Shao, J.; Han, M.; and Xu, Z. 2020. Large-Scale Multi-View Subspace Clustering in Linear Time. In *AAAI*, 4412–4419.

Li, X.; Sun, Y.; Sun, Q.; and Ren, Z. 2023a. Consensus Cluster Center Guided Latent Multi-Kernel Clustering. *IEEE TCSVT*, 33(6): 2864–2876.

Li, X.; Sun, Y.; Sun, Q.; Ren, Z.; and Sun, Y. 2023b. Cross-view Graph Matching Guided Anchor Alignment for Incomplete Multi-view Clustering. *Inf. Fusion*, 100: 101941.

Li, X.; Zhang, H.; Wang, R.; and Nie, F. 2022. Multiview Clustering: A Scalable and Parameter-Free Bipartite Graph Fusion Method. *IEEE TPAMI*, 44(1): 330–344.

Li, Y.; Nie, F.; Huang, H.; and Huang, J. 2015. Large-Scale Multi-View Spectral Clustering via Bipartite Graph. In *AAAI*, 2750–2756.

Li, Y.; Yang, M.; and Zhang, Z. 2019. A Survey of Multi-View Representation Learning. *IEEE TKDE*, 31(10): 1863–1883.

Lin, Y.; Gou, Y.; Liu, Z.; Li, B.; Lv, J.; and Peng, X. 2021. COMPLETER: Incomplete Multi-View Clustering via Contrastive Prediction. In *CVPR*, 11174–11183.

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 *ICCV*, 12324–12333.

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*, 7576–7584.

Liu, X. 2023. SimpleMKKM: Simple Multiple Kernel K-Means. *IEEE TPAMI*, 45(4): 5174–5186.

Lu, C.; Yan, S.; and Lin, Z. 2016. Convex Sparse Spectral Clustering: Single-View to Multi-View. *IEEE TIP*, 25(6): 2833–2843.

Luo, S.; Zhang, C.; Zhang, W.; and Cao, X. 2018. Consistent and Specific Multi-View Subspace Clustering. In *AAAI*, 3730–3737.

Lv, W.; Zhang, C.; Li, H.; Jia, X.; and Chen, C. 2023. Joint Projection Learning and Tensor Decomposition-Based Incomplete Multiview Clustering. *IEEE TNNLS*.

Qiang, Q.; Zhang, B.; Wang, F.; and Nie, F. 2021. Fast Multi-view Discrete Clustering with Anchor Graphs. In *AAAI*, 9360–9367.

Sun, M.; Zhang, P.; Wang, S.; Zhou, S.; Tu, W.; Liu, X.; Zhu, E.; and Wang, C. 2021. Scalable Multi-view Subspace Clustering with Unified Anchors. In *ACM MM*, 3528–3536.

Sun, Y.; Ren, Z.; Hu, P.; Peng, D.; and Wang, X. 2023. Hierarchical Consensus Hashing for Cross-Modal Retrieval. *IEEE TMM*.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *JMLR*, 9(11).

Wan, X.; Liu, J.; Liang, W.; Liu, X.; Wen, Y.; and Zhu, E. 2022. Continual Multi-view Clustering. In *ACM MM*, 3676–3684.

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*.

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

Wen, J.; Xu, G.; Liu, C.; Fei, L.; Huang, C.; Wang, W.; and Xu, Y. 2023a. Localized and Balanced Efficient Incomplete Multi-view Clustering. In *ACM MM*, 2927–2935.

Wen, J.; Xu, G.; Tang, Z.; Wang, W.; Fei, L.; and Xu, Y. 2023b. Graph Regularized and Feature Aware Matrix Factorization for Robust Incomplete Multi-view Clustering. *IEEE TCSVT*.

Wen, J.; Zhang, Z.; Fei, L.; Zhang, B.; Xu, Y.; Zhang, Z.; and Li, J. 2023c. A Survey on Incomplete Multiview Clustering. *IEEE Trans. Syst. Man Cybern. Syst.*, 53(2): 1136–1149.

Wen, J.; Zhang, Z.; Xu, Y.; Zhang, B.; Fei, L.; and Xie, G. 2020. CDIMC-net: Cognitive Deep Incomplete Multi-view Clustering Network. In *IJCAI*, 3230–3236.

Wen, Z.; and Yin, W. 2013. A feasible method for optimization with orthogonality constraints. *Math. Program.*, 142(1-2): 397–434.

Wu, J.; Lin, Z.; and Zha, H. 2019. Essential Tensor Learning for Multi-View Spectral Clustering. *IEEE TIP*, 28(12): 5910–5922.

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

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

Xie, Y.; Tao, D.; Zhang, W.; Liu, Y.; Zhang, L.; and Qu, Y. 2018. On Unifying Multi-view Self-Representations for Clustering by Tensor Multi-rank Minimization. *IJCV*, 126(11): 1157–1179.

Xu, J.; Tang, H.; Ren, Y.; Peng, L.; Zhu, X.; and He, L. 2022. Multi-level Feature Learning for Contrastive Multi-view Clustering. In *CVPR*, 16030–16039.

Yan, W.; Zhang, Y.; Lv, C.; Tang, C.; Yue, G.; Liao, L.; and Lin, W. 2023. GCFAgg: Global and Cross-View Feature Aggregation for Multi-View Clustering. In *CVPR*, 19863–19872.

Yang, B.; Zhang, X.; Li, Z.; Nie, F.; and Wang, F. 2023. Efficient Multi-View K-Means Clustering With Multiple Anchor Graphs. *IEEE TKDE*, 35(7): 6887–6900.

Zhang, C.; Hu, Q.; Fu, H.; Zhu, P.; and Cao, X. 2017. Latent Multi-view Subspace Clustering. In *CVPR*, 4333–4341.

Zhang, C.; Li, H.; Chen, C.; Jia, X.; and Chen, C. 2022. Low-Rank Tensor Regularized Views Recovery for Incomplete Multiview Clustering. *IEEE TNNLS*.

Zhang, C.; Li, H.; Gao, Y.; and Chen, C. 2023a. Weakly-Supervised Enhanced Semantic-Aware Hashing for Cross-Modal Retrieval. *IEEE TKDE*, 35(6): 6475–6488.

Zhang, C.; Li, H.; Lv, W.; Huang, Z.; Gao, Y.; and Chen, C. 2023b. Enhanced Tensor Low-Rank and Sparse Representation Recovery for Incomplete Multi-View Clustering. In *AAAI*, 11174–11182.

Zhang, C.; Wei, J.; Wang, B.; Li, Z.; Chen, C.; and Li, H. 2023c. Robust Spectral Embedding Completion Based Incomplete Multi-view Clustering. In *ACM MM*, 300–308.

Zhang, P.; Wang, S.; Li, L.; Zhang, C.; Liu, X.; Zhu, E.; Liu, Z.; Zhou, L.; and Luo, L. 2023d. Let the Data Choose: Flexible and Diverse Anchor Graph Fusion for Scalable Multi-View Clustering. In *AAAI*, volume 37, 11262–11269.