Continual Multi-View Clustering with Consistent Anchor Guidance

Chao Zhang[1] , **Deng Xu**[1] , **Xiuyi Jia**[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, dengxu}@smail.nju.edu.cn, jiaxy@njust.edu.cn, {clchen, huaxiongli}@nju.edu.cn

Abstract

Multi-view clustering (MVC) has recently attracted much attention. Most existing approaches are designed for fixed multi-view data, and cannot deal with the common streaming data in real world. In this paper, we address this problem by proposing a consistent Anchor guided Continual MVC (ACMVC) method in a two-stage way. In initial learning stage, a low-rank anchor graph based model is constructed. In continual learning stage, to leverage the historical knowledge, the multi-level anchor information is reused to refine the model via adding consistency regularization. It not only provides prior knowledge to enhance the exploration on current data, but also captures the similarity relationship between previous and current data, enabling a comprehensive exploitation on streaming data. The proposed model can be optimized efficiently with linear time and space complexity. Experiments demonstrate the effectiveness and efficiency of our method compared with some state-of-the-art approaches.

1 Introduction

In real-world scenarios, data are usually gathered from diverse sources, which form multi-view data and show the nature heterogeneity across views. Learning from such data has become a hot research topic due to the surge of open-world problems entailing multi-view data [Sun *et al.*, 2024b; Jiang *et al.*, 2023; Jiang *et al.*, 2024; Li *et al.*, 2023a; Zhang *et al.*, 2023a; Sun *et al.*, 2024a; Sun *et al.*, 2023], especially in unsupervised cases. Clustering aims to divide data into disjoint groups, which is a fundamental unsupervised task in many applications. Focusing on discovering the cross-view consistency and complementarity for comprehensive data utilization, many multi-view clustering (MVC) approaches are developed, such as subspace based methods [Kang *et al.*, 2020; Chen *et al.*, 2022; Qin *et al.*, 2024], kernel based methods [Liu *et al.*, 2021b; Ren *et al.*, 2021], graph based methods [Zhang *et al.*, 2023b; Li *et al.*, 2023b; Li *et al.*, 2022a], deep learning based methods [Huang *et al.*, 2019; Trosten *et al.*, 2021; Lin *et al.*, 2021; Cai *et al.*, 2022; Cai *et al.*, 2024], etc. In spite of the success of these approaches, most of them fall within static MVC approaches, *i.e.*, they can only handle the problem where the data are fixed, which is usually not satisfied in practice.

One more generalized issue is that data are continually collected and accumulated over time, constituting a dynamic data stream [Wang *et al.*, 2023; Lange *et al.*, 2022]. For example, on social media platforms like Twitter, thousands of images, texts and videos are posted every second from users, which form continuous multi-view data flow. In an industrial setting, the continuous collection of different sensors also represents a dynamic stream of multi-view data. Apparently, analyzing and clustering such dynamic data poses greater challenges than static data.

A straightforward approach to tackle this challenge is to store all historical and incoming data, and then apply traditional MVC methods on entire data for clustering. However, it will incur substantial storage and time overheads, and also waste the model knowledge obtained from previous data. Additionally, the gathered data may only be available temporarily due to privacy concerns. Continual learning (CL) [Lange *et al.*, 2022] provides an effective solution to learn from dynamic streaming data, which aims to gradually expand acquired knowledge and use it for future learning without re-training all previous data. Concentrating on the catastrophic forgetting issue, many CL methods are designed, which can be roughly divided into five categories [Wang *et al.*, 2023], including regularization-based, replay-based, optimization-based, representation-based, and architecture-based approaches. These various methods attempt to integrate new knowledge while retaining previous knowledge, enabling the model continually adapt and evolve over time. While CL has been widely studied [Li and Hoiem, 2018; Zhou *et al.*, 2022; Hou *et al.*, 2023; Li and Gu, 2023], its exploration of MVC remains relatively limited, leaving a vast space for further research and investigation.

In this paper, we propose a new consistent Anchor guided Continual MVC (ACMVC) method for streaming multi-view data clustering. ACMVC consists of two stages, *Initial learning* stage (I-stage) and *Continual learning* stage (C-stage). In I-stage, the model learns from existing data, which is the same as static MVC. In C-stage, a chunk of data arrives per round, and the model adapts by transferring historical

*Corresponding author.

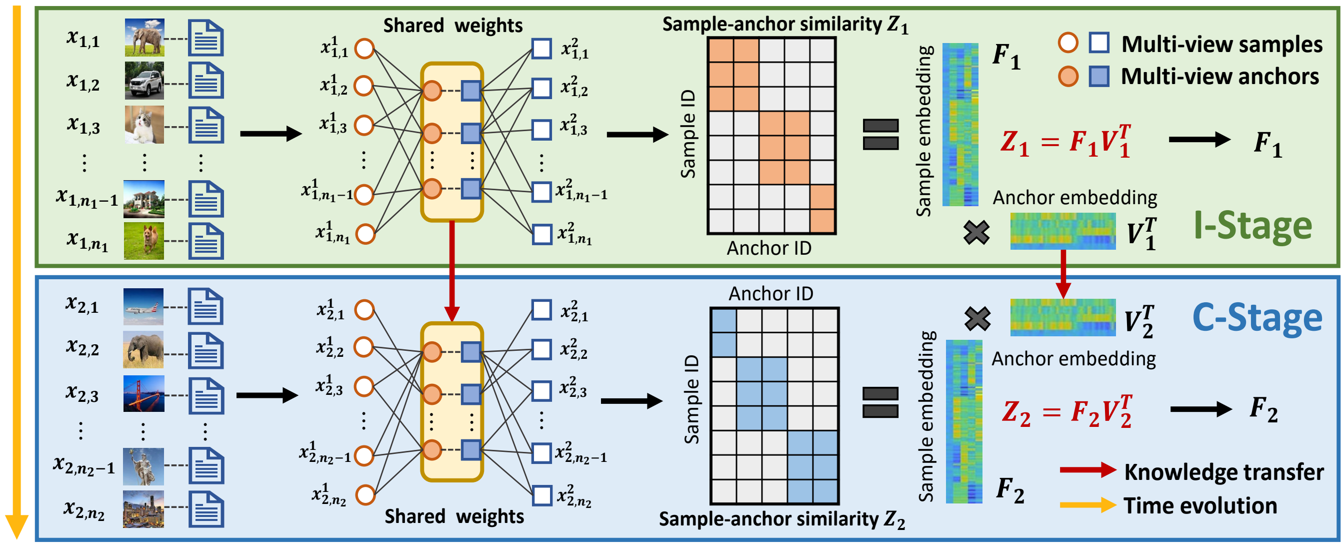

Figure 1: Framework of the proposed ACMVC method.

knowledge to explore new data. Figure 1 provides the framework of our method, showcasing the process using two data chunks. In I-stage, ACMVC adaptively learns some anchors for each view, and a shared non-negative sample-anchor similarity matrix. The matrix not only exploits the consistent and complementary information, but also aligns the multi-view anchors. The similarity matrix is further factorized into a low-dimensional sample embedding and an anchor embedding which preserves the similarity relationships and conversely enhances the low-rank property of similarity matrix. In C-stage, rather than re-learning all previous data, we refine the model by leveraging transferred anchor knowledge. We achieve it by imposing a regularization term on original anchors and latent anchor embeddings, which helps to capture cross-chunk consistency information and enhances the model's adaptability and performance. The main contributions of this work are summarized as follow:

- We propose a new continual multi-view clustering method, integrating multi-level anchor knowledge reuse. It can efficiently learn from streaming multi-view data with linear space and time complexity.
- We solve the problem via a two-stage strategy. In I-stage, a basic model is built with multi-level anchors and similarity learning. In C-stage, the model is refined by adding regularization to reuse anchor knowledge.
- We develop an efficient alternate algorithm to solve the model. Comparative experiments on popular datasets demonstrate the effectiveness and efficiency of our method in clustering streaming multi-view data.

2 Related Work

In this section, we present a brief overview of related research concerning multi-view clustering and continual learning.

Multi-view Clustering. During the past decade, numerous methods for MVC have been proposed, including graph-based, subspace-based, kernel-based, and deep learning-based approaches. Graph-based methods have gained significant popularity among these techniques [Huang *et al.*, 2022; Zhang *et al.*, 2023b; Zhang *et al.*, 2022a]. They typically involve constructing a similarity graph to delineate pairwise relationships between data points and subsequently applying classical spectral clustering framework [Tang *et al.*, 2019; Wang *et al.*, 2020]. Nevertheless, constructing a complete $n \times n$ graph makes them unapplicable to large-scale data. Thus, anchor graph-based technique is developed, focusing on constructing a concise sample-anchor bipartite graph [Li *et al.*, 2015; Zhang *et al.*, 2024]. This approach greatly reduces the time and space consumption while upholding considerable clustering performance. For example, [Kang *et al.*, 2020] adopt k-means to select representative anchors. [Li *et al.*, 2022b] propose a directly alternate sampling scheme by measuring feature score to select anchors. These methods use heuristic strategies to determine anchors. In contrast, some researchers take an adaptive approach to learn anchors and integrate it with anchor graph learning in a unified model. For example, [Sun *et al.*, 2021] learn a consensus anchor matrix and a non-negative graph for all views in a latent subspace. Although these methods have attained significant success, they generally belong to static MVC approaches, and cannot handle streaming data.

Continual Learning. CL studies the problem of learning from an infinite stream of data, aiming to progressively expanding existing knowledge and using it to facilitate future learning. It is also referred to as incremental learning, lifelong learning in most cases, without a strict distinction [Wang *et al.*, 2023]. CL encompasses a variety of dynamic scenarios such as instance-incremental, class-incremental, domain-incremental, task-incremental and more, most of which belong to supervised learning. Recently, some researchers introduced CL into unsupervised MVC, focusing primarily on two dynamic scenarios, view-incremental and instance-incremental. To tackle the view-incremental problem, [Yin *et al.*, 2021] learn a consensus low-rank and sparse similarity graph for existing data, and use it to guide new graph learning upon the arrival of additional views. [Wan *et al.*, 2022] keep a consensus partition matrix. To tackle the instance-incremental problem, [Shao *et al.*, 2016] extends the classical non-negative matrix factorization (NMF) to an online version with a consistent base matrix regularization. [Hu and Chen, 2019] apply matrix factorization to learn cluster indicators and a unified base matrix for knowledge reuse.

3 Proposed Method

In this section, we first give the notations and problem definition, and then delve into the details of I-stage and C-stage within our proposed ACMVC framework.

3.1 Notations and Problem Statement

Scalar, vector and matrix are denoted by lowercase, lowercase bold and uppercase bold letters (e.g., b, $\mathbf{b}$, $\mathbf{B}$), respectively. $\mathrm{Tr}(\cdot)$ returns the matrix trace. $\mathbf{1}$ is a vector whose all elements are 1, and $\mathbf{I}$ is an identity matrix.

Our method addresses the streaming multi-view data clustering problem. We assume a chunk of multi-view data arrives per round. Let $\mathcal{X}_t = \{\mathbf{X}_t^1, \mathbf{X}_t^2, ..., \mathbf{X}_t^m\}$ denote the received data in the t-th round with total m views, and $\mathbf{X}_t^v = [\mathbf{x}_{t,1}^v; \mathbf{x}_{t,2}^v; ...; \mathbf{x}_{t,n_t}^v] \in \mathbb{R}^{n_t \times d_v}$ denotes the v-th view feature matrix, n_t and d_v represent the sample number and feature dimension, respectively. In I-stage (round 1), we obtain data $\mathcal{X}_1$ and construct an initial model $\mathcal{H}_1$ for clustering. In C-stage (round $t, t \geq 2$), we receive new data $\mathcal{X}_t$ and adapt $\mathcal{H}_{t-1}$ to $\mathcal{H}_t$ by jointly acquiring new knowledge and transferring previous knowledge. The goal is to use the evolving model to partition all data into c clusters.

3.2 I-stage

During the I-Stage, we need to construct a basic model for $\mathcal{X}_1$, which not only effectively explores the relations among data for cluster structure discovery, but also offers easy adaptability to process streaming data. Inspired by the efficiency of anchor based approaches, we first develop the following primary model:

$$\begin{aligned} &\min_{\mathbf{w}_1, \mathbf{A}_1^v, \mathbf{Z}} \sum_{v=1}^{m} \frac{1}{w_1^v} \|\mathbf{X}_1^v - \mathbf{Z}_1 \mathbf{A}_1^v\|_F^2 \\ &s.t.\ \mathbf{Z}_1 \geq 0, \mathbf{A}_1^v \mathbf{A}_1^{vT} = \mathbf{I}, \mathbf{w}_1^T \mathbf{1} = 1, \mathbf{w}_1 \geq 0. \end{aligned} \tag{1}$$

where $\mathbf{A}_1^v \in \mathbb{R}^{k \times d_v}$ denotes the anchor matrix containing k anchors, $\mathbf{Z}_1 \in \mathbb{R}^{n_1 \times k}$ is a shared sample-anchor similarity matrix indicating the similarity relationship between each sample and anchors, $\mathbf{w}_1 = [w_1^1, w_1^2, ..., w_1^m]^T \in \mathbb{R}^m$ is a view weight vector. We employ orthogonal constraint on $\mathbf{A}_1^v$ to ensure diverse anchors and avoid arbitrary scaling. The non-negative constraint on $\mathbf{Z}$ makes the learned similarity matrix physically meaningful.

For a clustering-friendly similarity learning, it is expected that the similarity matrix $\mathbf{Z}$ is low-rank with $\mathrm{rank}(\mathbf{Z}) = c$, since the data are approximately drawn from c subspaces [Liu *et al.*, 2013; Zhang *et al.*, 2022b]. However, it is hard to optimize by explicitly enforcing the the rank to be c. Inspired by matrix factorization, we factorize the assymetric similarity graph $\mathbf{Z}$ into two latent factor embeddings, and address the following problem:

$$\begin{aligned} &\min_{\substack{\mathbf{w}_1, \mathbf{A}_1^v, \mathbf{Z}_1 \\ \mathbf{F}_1, \mathbf{V}_1}} \sum_{v=1}^{m} \frac{1}{w_1^v} \|\mathbf{X}_1^v - \mathbf{Z}_1 \mathbf{A}_1^v\|_F^2 + \alpha \|\mathbf{Z}_1 - \mathbf{F}_1 \mathbf{V}_1^T\|_F^2 \\ &s.t. \begin{cases} \mathbf{A}_1^v \mathbf{A}_1^{vT} = \mathbf{I}, \mathbf{F}_1^T \mathbf{F}_1 = \mathbf{I}, \mathbf{V}_1^T \mathbf{V}_1 = \mathbf{I}, \\ \mathbf{Z}_1 \geq 0, \mathbf{w}_1^T \mathbf{1} = 1, \mathbf{w}_1 \geq 0, \end{cases} \end{aligned} \tag{2}$$

where $\mathbf{F}_1 \in \mathbb{R}^{n_1 \times c}$ and $\mathbf{V}_1 \in \mathbb{R}^{m \times c}$ can be viewed as sample embedding and anchor embedding, respectively, which encode the latent information of samples and anchors. The sample-anchor similarities are preserved into latent embeddings by inner product. Eq. (2) embodies an adaptive learning process that dynamically seeks representative anchors and constructs a low-rank similarity graph. The similarities are preserved into latent representation, where the sample embeddings can be used for k-means clustering.

3.3 C-stage

When a new chunk of data $\mathcal{X}_t$ arrives at the t-th ($t \geq 2$) round, it is essential to learn the latent sample embedding $\mathbf{F}_t$ like that in I-stage. Moreover, $\mathbf{F}_t$ should not only discern within-chunk similarities but also capture the inter-chunk data similarities. Based on the fundamental model in I-stage, a straightforward way is to use the same anchors and latent anchor embeddings to guide new similarity construction and sample embedding learning, whose objective function can be described as

$$\begin{aligned} &\min_{\mathbf{w}_t, \mathbf{Z}_t, \mathbf{F}_t} \sum_{v=1}^{m} \frac{1}{w_t^v} \|\mathbf{X}_t^v - \mathbf{Z}_t \mathbf{A}_{t-1}^v\|_F^2 + \alpha \|\mathbf{Z}_t - \mathbf{F}_t \mathbf{V}_{t-1}^T\|_F^2 \\ &s.t.\ \mathbf{Z}_t \geq 0, \mathbf{F}_t^T \mathbf{F}_t = \mathbf{I}, \mathbf{w}_t^T \mathbf{1} = 1, \mathbf{w}_t \geq 0, \end{aligned} \tag{3}$$

where $\{\mathbf{A}_{t-1}^v\}_{v=1}^m$ and $\mathbf{V}_{t-1}$ are directly inherited from the historical model $\mathcal{H}_{t-1}$, and it has $\mathbf{A}_{t-1}^v = \mathbf{A}_{t-2}^v = ... = \mathbf{A}_1^v$ and $\mathbf{V}_{t-1} = \mathbf{V}_{t-2} = ... = \mathbf{V}_1$.

However, with the continual increase and distribution shift of data, the historical anchor information may be sub-optimal to guide the similarity graph and sample embeddings learning of new data. Thus, we plan to refine the anchor information while maintaining their semantic consistency across chunks, which can be achieved by

$$\|\mathbf{A}_t^v - \mathbf{A}_{t-1}^v\|_F^2 \leq \varepsilon_1, \text{and } \|\mathbf{V}_t - \mathbf{V}_{t-1}\|_F^2 \leq \varepsilon_2, \tag{4}$$

where $\varepsilon_1, \varepsilon_2 \geq 0$ are tolerance error. For ease of optimization and avoid intricate parameter adjustment, we formulate the objective function in C-stage as follows:

$$\begin{aligned} &\min_{\substack{\mathbf{w}_t, \mathbf{A}_t^v, \mathbf{Z}_t \\ \mathbf{F}_t, \mathbf{V}_t}} \sum_{v=1}^{m} \frac{1}{w_t^v} \|\mathbf{X}_t^v - \mathbf{Z}_t \mathbf{A}_t^v\|_F^2 + \alpha \|\mathbf{Z}_t - \mathbf{F}_t \mathbf{V}_t^T\|_F^2 \\ &+ \beta \left(\sum_{v=1}^{m} \|\mathbf{A}_t^v - \mathbf{A}_{t-1}^v\|_F^2 + \|\mathbf{V}_t - \mathbf{V}_{t-1}\|_F^2 \right) \\ &s.t. \begin{cases} \mathbf{A}_t^v \mathbf{A}_t^{vT} = \mathbf{I}, \mathbf{F}_t^T \mathbf{F}_t = \mathbf{I}, \mathbf{V}_t^T \mathbf{V}_t = \mathbf{I}, \\ \mathbf{Z}_t \geq 0, \mathbf{w}_t^T \mathbf{1} = 1, \mathbf{w}_t \geq 0. \end{cases} \end{aligned} \tag{5}$$

When $\beta \to \infty$, Eq. (5) is equivalent to Eq. (3). When $\beta = 0$, Eq. (5) is independent with historical models and no previous knowledge is used. Eq. (5) extends Eq. (2) by adding regularization on multi-view anchors $\{\mathbf{A}_t^v\}_{v=1}^m$ and anchor embedding $\mathbf{V}_t$. With the guidance of consistent multi-level anchor information, on the one hand, our model leverages historical knowledge to enhance the exploration on new data. On the other hand, the similarities of data in different chunks are also mined to derive homogeneous latent embeddings, enabling a comprehensive exploitation for streaming data.

4 Algorithm

In this section, we present the optimization algorithm to solve the proposed model, and analyze the algorithm complexity.

4.1 Optimization Algorithm

We can observe that, when $\beta = 0$, Eq. (5) is exactly Eq. (2), and thus, we only need to optimize the former. We use alternate search strategy to solve it.

$\mathbf{A}_t^v$ Step: Because the anchors of different views are independent, we can obtain each view-specific anchor matrix by solving the following problem:

$$\min_{\mathbf{A}_t^v \mathbf{A}_t^{vT} = \mathbf{I}} \frac{1}{w_t^v} \|\mathbf{X}_t^v - \mathbf{Z}_t \mathbf{A}_t^v\|_F^2 + \beta \|\mathbf{A}_t^v - \mathbf{A}_{t-1}^v\|_F^2. \quad (6)$$

By utilizing $\mathbf{A}_t^v \mathbf{A}_t^{vT} = \mathbf{A}_{t-1}^v (\mathbf{A}_{t-1}^v)^T = \mathbf{I}$, we can convert Eq. (6) as the following maximum optimization problem:

$$\max_{\mathbf{A}_t^v} \operatorname{Tr}\left(\mathbf{A}_t^{vT} \mathbf{Q}_t^v\right) \quad s.t.\ \mathbf{A}_t^v \mathbf{A}_t^{vT} = \mathbf{I}, \quad (7)$$

where $\mathbf{Q}_t^v = \frac{1}{w_t^v} \mathbf{Z}_t^T \mathbf{X}_t^v + \beta \mathbf{A}_{t-1}^v$. This problem can be solved by the following theorem.

Theorem 1. *Suppose the singular value decomposition (SVD) of $\mathbf{P} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, the optimal solution to*

$$\max_{\mathbf{G}} \operatorname{Tr}(\mathbf{G}\mathbf{P}) \quad s.t.\ \mathbf{G}\mathbf{G}^T = \mathbf{I}, \quad (8)$$

is given by $\mathbf{G}^* = \mathbf{V}\mathbf{U}^T$.

Proof. By introducing $\mathbf{P} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ into Eq. (8), we have

$$\begin{aligned} \operatorname{tr}(\mathbf{G}\mathbf{P}) &= \operatorname{tr}\left(\mathbf{G}\mathbf{U}\mathbf{\Sigma}(\mathbf{V})^T\right) = \operatorname{tr}\left(\mathbf{\Sigma}\mathbf{V}^T\mathbf{G}\mathbf{U}\right) \\ &= \operatorname{tr}(\mathbf{\Sigma}\mathbf{M}) = \sum_i s_{ii} m_{ii}, \end{aligned} \quad (9)$$

where $\mathbf{M} = \mathbf{V}^T\mathbf{G}\mathbf{U}$. It can be easily verified that $\mathbf{M}^T\mathbf{M} = \mathbf{I}$. Thus, we have $-1 \le m_{ii} \le 1$, and then

$$\operatorname{tr}(\mathbf{G}\mathbf{P}) = \sum_i s_{ii} m_{ii} \leqslant \sum_i s_{ii}. \quad (10)$$

Without of generality, let $\mathbf{M} = \mathbf{V}^T\mathbf{G}\mathbf{U} = \mathbf{I}$, i.e., $\mathbf{G}^* = \mathbf{V}\mathbf{U}^T$, $\operatorname{Tr}(\mathbf{G}\mathbf{P})$ gets the maximum. This completes the proof. □

$\mathbf{Z}_t$ Step: When other variables are fixed, the optimization problme of $\mathbf{Z}_t$ becomes

$$\min_{\mathbf{Z}_t \ge 0} \sum_{v=1}^m \frac{1}{w_t^v} \|\mathbf{X}_t^v - \mathbf{Z}_t \mathbf{A}_t^v\|_F^2 + \alpha \|\mathbf{Z}_t - \mathbf{F}_t \mathbf{V}_t^T\|_F^2. \quad (11)$$

Algorithm 1 ACMVC algorithm

Input: Multi-view data stream $\{\mathcal{X}_1, \mathcal{X}_2, ..., \mathcal{X}_t, ...\}$, cluster number c, anchor number k, model parameters α, β.
Output: Apply k-means on $\{\mathbf{F}_1, \mathbf{F}_2, ..., \mathbf{F}_t, ...\}$.

for $t = 1; t++$ **do**
  Initialize $\mathbf{A}_t^v, \mathbf{Z}_t, \mathbf{F}_t, \mathbf{V}_t$ by $\mathbf{0}$, and $w_t^v = 1/m$.
  while not converged **do**
    Update $\{\mathbf{A}_t^v\}_{v=1}^m$ by Thm. 1; ▷ $\beta = 0$ if $t = 1$
    Update $\mathbf{Z}_t$ by Eq. (14);
    Update $\mathbf{F}_t$ by solving (15);
    Update $\mathbf{V}_t$ by solving (16); ▷ $\beta = 0$ if $t = 1$
    Update $\{w_t^v\}_{v=1}^m$ by Eq. (19);
  end while
end for

Because $\mathbf{A}_t^v \mathbf{A}_t^{vT} = \mathbf{I}$, with some simple operation, we can rewrite the problem as follows:

$$\min_{\mathbf{Z}_t \ge 0} \sum_{v=1}^m \frac{1}{w_t^v} \|\mathbf{X}_t^v \mathbf{A}_t^{vT} - \mathbf{Z}_t\|_F^2 + \alpha \|\mathbf{Z}_t - \mathbf{F}_t \mathbf{V}_t^T\|_F^2. \quad (12)$$

It can be further simplified as

$$\min_{\mathbf{Z}_t} \|\mathbf{P}_t - \mathbf{Z}_t\|_F^2 \quad s.t.\ \mathbf{Z}_t \ge 0, \quad (13)$$

where $\mathbf{P}_t = \frac{\alpha \mathbf{F}_t \mathbf{V}_t^T + \sum_{v=1}^m \mathbf{X}_t^v \mathbf{A}_t^{vT}/w_t^v}{\alpha + \sum_{v=1}^m 1/w_t^v}$. Obviously, the optimal solution of $\mathbf{Z}_t$ is given by

$$\mathbf{Z}_t = \max(\mathbf{P}_t, 0). \quad (14)$$

$\mathbf{F}_t$ Step: With some manipulation, we can rewrite the problem w.r.t. $\mathbf{F}_t$ as follows:

$$\max_{\mathbf{F}_t} \operatorname{Tr}\left(\mathbf{F}_t^T \mathbf{Z}_t \mathbf{V}_t^T\right) \quad s.t.\ \mathbf{F}_t^T \mathbf{F}_t = \mathbf{I}. \quad (15)$$

Similar to $\mathbf{A}_t^v$, it can also be solved by Theorem 1.

$\mathbf{V}_t$ Step: The problem w.r.t. $\mathbf{V}_t$ is

$$\max_{\mathbf{V}_t} \operatorname{Tr}\left(\mathbf{V}_t^T \left(\alpha \mathbf{Z}_t^T \mathbf{F}_t + \beta \mathbf{V}_{t-1}\right)\right) \quad s.t.\ \mathbf{V}_t^T \mathbf{V}_t = \mathbf{I}. \quad (16)$$

We also use SVD to resolve it.

$\mathbf{w}_t$ Step: Defining $d_t^v = \|\mathbf{X}_t^v - \mathbf{Z}_t \mathbf{A}_t^v\|_F$, the optimization problem w.r.t. $\mathbf{w}_t$ is

$$\min_{\mathbf{w}_t} \sum_{v=1}^m \frac{(d_t^v)^2}{w_t^v} \quad s.t.\ \sum_v w_t^v = 1, w_t^v \ge 0. \quad (17)$$

According to Cauchy-Schwartz inequality, we get

$$\sum_{v=1}^m \frac{(d_t^v)^2}{w_t^v} \overset{\text{(i)}}{=} \left(\sum_{v=1}^m \frac{(d_t^v)^2}{w_t^v}\right)\left(\sum_{v=1}^m w_t^v\right) \overset{\text{(ii)}}{\geqslant} \left(\sum_{v=1}^m d_t^v\right)^2, \quad (18)$$

where (i) holds because $\sum_v w_t^v = 1$, and (ii) holds because $\sqrt{w_t^v} \propto \frac{d_t^v}{\sqrt{w_t^v}}$. The optimal view weight is

$$w_t^v = d_t^v / \sum_{v=1}^m d_t^v. \quad (19)$$

By iteratively executing the above update steps, the objective function can be minimized until convergence. For convenience, we summarize the entire optimization process in Algorithm 1.

4.2 Complexity Analysis

Time Complexity. When updating $\mathbf{A}_t^v$, the main time costs come from the matrix multiplication and SVD operation, which take $O(kd_vn_t)$ and $O(k^2n_t)$ complexity, respectively. When updating $\mathbf{Z}_t$, its complexity is $O((c+d)kn_t)$ with $d = \sum_{v=1}^{m} d_v$. For $\mathbf{F}_t$ and $\mathbf{V}_t$, the complexity is $O(ckn_t + c^2n_t)$ and $O(ckn_t + c^2k)$, respectively. The time complexity for updating $\mathbf{w}_t$ is $O(kdn_t)$. Because $k, c \ll n_t$, the overall time complexity of our algorithm is $O(dn_t)$, which is linear to the number of samples.

Space Complexity. The major memory costs of our algorithm are variables $\mathbf{A}_t^v \in \mathbb{R}^{k\times d_v}$, $\mathbf{Z}_t \in \mathbb{R}^{n_t\times k}$, $\mathbf{F}_t \in \mathbb{R}^{n_t\times c}$, $\mathbf{V}_t \in \mathbb{R}^{k\times c}$. Thus, the space complexity of our method is also linear to the number of samples.

5 Experiment

In this section, we conduct experiments on several popular multi-view datasets to investigate the effectiveness of our method and compare it with some state-of-the-art approaches. We also evaluate the convergence property, running time, parameter sensitivity, and ablation study of our model. The computing platform is MATLAB R2019b with Win10 System, Intel Core i7-8700 CPU@3.2GHz and 16GB RAM.

5.1 Experimental Setup

Datasets. We adopt seven popular multi-view datasets for experiments, including BBCSport[1], Mfeat[2], Wiki[3], MITIndoor[4], Caltech101[5], Fashion[6], and VGGFace[7] datasets. Table 1 lists the general statistics.

Baselines. We compare ACMVC with following baselines:

- **SMVSC** [Sun *et al.*, 2021] adaptively learns unified latent anchors, and a consensus subspace representation.
- **OPMC** [Liu *et al.*, 2021a] integrates matrix factorization and discrete cluster indicator matrix learning into a unified model.
- **OMSC** [Chen *et al.*, 2022] jointly considers anchor learning, anchor graph construction and discrete partition for efficient MVC.
- **EOMSC** [Liu *et al.*, 2022] jointly learns anchors and low-rank anchor graph with a connectivity constraint.
- **AWMVC** [Wan *et al.*, 2023] learns multiple subspace representations in different dimensions for each view, and then fuses them in an auto-weighted ensemble way.
- **OMVC** [Shao *et al.*, 2016] is an online MVC method, which learns the latent representations by weighted NMF and pushes them towards a consensus one.

Dataset	Sample	View	Cluster	Feature
BBCSport	544	2	5	3183, 3203
Mfeat	2000	6	10	216, 76, 64, 6, 240, 47
Wiki	2688	2	10	128, 10
MITIndoor	5360	4	67	4096, 3600, 1770, 1240
Caltech101	9144	5	102	48, 40, 254, 512, 928
Fashion	10000	3	10	784, 784, 784
VGGFace	34027	4	50	944, 576, 512, 640

Table 1: General statistics of datasets.

- **OPIMC** [Hu and Chen, 2019] is also an online method, which directly learns a consensus discrete partition matrix by weighted matrix factorization.

Implementation Details. Our method addresses the streaming multi-view data clustering problem. To construct the streaming data, we split the whole dataset into several chunks, and a chunk of data arrives in each round. The chunk size is set to 100 for BBCSport, 1000 for Mfeat, Wiki, MITIndoor, Caltech101, and 5000 for Fashion and VGGFace. The source codes of baselines are publicly available, and we search their optimal hyper-parameters following the suggestions of original papers. For our ACMVC, we search the parameters α and β in the range of $\{10^0, 10^1, ..., 10^6\}$, and fix the number of anchors $k = 5c$ in all experiments for simplicity, where c is the number of clusters. We adopt five popular clustering evaluation metrics, i.e., accuracy (ACC), normalized mutual information (NMI), purity (PUR), adjusted rand index (ARI) and Fscore. The larger values indicate the better clustering performance. We repeat each experiment 10 times to report the mean values as well as standard deviation.

5.2 Clustering Results

Table 2 lists the clustering results of our method and baselines on seven datasets w.r.t. five metrics, where the best results and second-best are highlighted in red and blue, respectively. We have the following observations:

- Our method generally achieves the best or second-best results on all datasets. EOMSC and AWMVC are two advanced offline MVC algorithm. They first accumulate each data chunk, and then perform clustering on the accumulated data, which can comprehensively explore the similarity relations among all data. Thus, they also obtain the best and competitive performance in some cases. Although our method sequentially learns from each data chunk in an online manner, it still gets excellent results and outperforms the offline methods like SMVSC, OPMC, OMSC and EOMSC in most cases, demonstrating the effectiveness of our consistent anchor based continual learning mechanism.
- OMVC and OPIMC are two continual learning methods based on multi-view matrix factorization. The performance of OMVC is not desirable, and in most cases, it is inferior to some offline MVC approaches like OMSC, EOMSC, and AWMVC. OPIMC generally achieves better results than OMVC, and it obtains competitive performance on Wiki, Caltech101 and Fashion datasets.

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

[2] https://archive.ics.uci.edu/dataset/72/multiple+features

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

[4] https://web.mit.edu/torralba/www/indoor.html

[5] https://www.vision.caltech.edu/datasets/

[6] https://www.kaggle.com/datasets/paramaggarwal/fashion-product-images-dataset

[7] https://www.robots.ox.ac.uk/~vgg/data/vgg_face2/

	Metrics	SMVSC	OPMC	OMSC	EOMSC	AWMVC	OMVC	OPIMC	Ours
BBCSport	ACC	42.70(2.10)	53.79(5.12)	44.85(0.00)	41.54(0.00)	78.27(5.06)	69.19(2.81)	57.90(0.00)	80.68(0.10)
	NMI	16.19(2.04)	38.57(5.32)	19.92(0.00)	24.00(0.00)	67.33(2.34)	53.03(1.40)	32.47(0.00)	58.09(0.10)
	PUR	46.84(3.27)	62.21(4.65)	51.10(0.00)	57.54(0.00)	81.78(2.23)	73.16(0.78)	61.58(0.00)	80.68(0.10)
	ARI	10.23(2.39)	20.90(3.07)	13.73(0.00)	14.65(0.00)	62.64(1.94)	43.72(1.12)	28.25(0.00)	59.16(0.11)
	Fscore	32.70(1.95)	42.51(2.88)	35.59(0.00)	33.60(0.00)	70.95(1.48)	56.48(0.86)	46.57(0.00)	69.09(0.08)
Mfeat	ACC	68.66(8.61)	76.25(5.40)	80.80(0.00)	76.00(0.00)	70.72(5.32)	55.54(3.07)	80.60(0.00)	86.18(0.92)
	NMI	72.76(4.05)	77.06(3.76)	77.42(0.00)	82.08(0.00)	68.48(2.24)	56.86(1.30)	79.37(0.00)	78.29(0.68)
	PUR	68.87(8.47)	79.16(4.74)	80.80(0.00)	76.20(0.00)	73.01(4.04)	60.39(2.22)	81.25(0.00)	86.18(0.92)
	ARI	60.64(7.59)	68.99(6.22)	70.27(0.00)	69.94(0.00)	58.30(4.39)	43.22(2.51)	72.84(0.00)	73.36(1.31)
	Fscore	65.32(6.40)	72.25(5.99)	73.42(0.00)	73.42(0.00)	62.56(3.89)	49.25(2.61)	75.69(0.00)	76.03(1.17)
Wiki	ACC	30.51(0.96)	18.89(0.52)	38.59(0.00)	54.82(0.00)	17.41(6.12)	45.62(1.05)	53.99(0.00)	55.10(1.67)
	NMI	15.76(1.00)	6.01(0.11)	25.00(0.00)	53.26(0.00)	4.40(1.55)	32.52(0.29)	51.82(0.00)	52.96(0.20)
	PUR	34.35(0.88)	22.04(0.44)	41.24(0.00)	61.48(0.00)	19.53(6.86)	48.74(0.55)	58.33(0.00)	61.36(0.33)
	ARI	10.13(0.58)	2.78(0.18)	18.68(0.00)	42.88(0.00)	2.24(0.79)	25.62(0.69)	41.52(0.00)	42.46(1.25)
	Fscore	20.26(0.55)	13.42(0.15)	27.49(0.00)	48.93(0.00)	11.38(4.00)	33.63(0.51)	47.78(0.00)	48.58(1.17)
MITIndoor	ACC	19.50(0.40)	30.13(1.42)	20.41(0.00)	15.06(0.00)	42.98(1.58)	14.08(0.65)	21.23(0.00)	46.68(1.25)
	NMI	34.91(0.19)	45.00(0.39)	38.96(0.00)	28.43(0.00)	56.39(0.79)	26.56(0.40)	36.82(0.00)	58.74(0.76)
	PUR	20.16(0.43)	31.80(0.88)	21.19(0.00)	15.60(0.00)	44.82(1.39)	15.02(0.67)	22.72(0.00)	48.71(1.14)
	ARI	8.91(0.13)	15.84(0.43)	8.57(0.00)	5.29(0.00)	29.76(1.01)	3.23(0.37)	9.66(0.00)	31.99(0.94)
	Fscore	10.82(0.11)	17.26(0.41)	10.74(0.00)	7.62(0.00)	30.83(0.99)	5.13(0.29)	11.22(0.00)	33.03(0.93)
Caltech101	ACC	28.37(0.78)	23.61(0.52)	30.35(0.00)	22.19(0.00)	24.89(1.37)	13.02(0.41)	24.43(0.00)	26.46(0.95)
	NMI	34.68(0.64)	43.80(0.38)	36.67(0.00)	25.33(0.00)	48.61(0.68)	30.69(0.21)	33.89(0.00)	48.82(0.33)
	PUR	33.22(0.45)	42.17(0.43)	34.83(0.00)	26.01(0.00)	46.53(0.95)	28.44(0.54)	32.24(0.00)	47.05(0.47)
	ARI	15.40(0.82)	22.47(1.85)	15.23(0.00)	8.60(0.00)	19.41(1.58)	7.05(0.37)	17.94(0.00)	19.62(0.98)
	Fscore	19.11(0.74)	23.89(1.82)	19.04(0.00)	12.87(0.00)	20.66(1.57)	8.55(0.37)	21.01(0.00)	20.84(0.98)
Fashion	ACC	65.19(5.01)	65.74(6.09)	71.54(0.00)	67.73(0.00)	52.39(1.04)	62.33(4.95)	73.03(0.00)	75.77(0.05)
	NMI	72.79(1.94)	68.62(2.50)	76.14(0.00)	76.24(0.00)	56.31(1.92)	67.13(2.47)	76.40(0.00)	75.50(0.02)
	PUR	65.78(4.70)	62.86(5.24)	75.09(0.00)	67.87(0.00)	54.27(1.47)	65.97(4.44)	78.79(0.00)	80.15(0.02)
	ARI	60.38(3.61)	54.89(3.72)	62.55(0.00)	59.31(0.00)	44.78(1.08)	53.33(3.46)	63.13(0.00)	65.98(0.03)
	Fscore	64.94(2.97)	59.82(3.22)	66.82(0.00)	64.43(0.00)	49.49(1.62)	58.33(0.30)	67.60(0.00)	69.51(0.03)
VGGFace	ACC	9.92(0.17)	11.77(0.52)	11.21(0.00)	11.46(0.00)	13.47(0.52)	9.08(0.28)	10.81(0.00)	13.74(0.38)
	NMI	11.87(0.24)	14.39(0.53)	14.28(0.00)	14.14(0.00)	16.35(0.41)	10.98(0.11)	14.13(0.00)	15.77(0.33)
	PUR	10.33(0.19)	12.87(0.56)	11.87(0.00)	11.88(0.00)	14.53(0.52)	10.04(0.27)	11.81(0.00)	14.71(0.44)
	ARI	2.27(0.04)	3.72(0.20)	3.27(0.00)	3.21(0.00)	6.73(0.25)	4.24(0.08)	5.29(0.00)	6.70(0.15)
	Fscore	5.34(0.03)	5.72(0.19)	5.99(0.00)	5.86(0.00)	6.68(0.25)	4.17(0.08)	5.12(0.00)	6.62(0.15)

Table 2: Clustering results (%) w.r.t. five metrics on seven datasets.

Compared with the two methods, our approach further improves the clustering performance, implying that the adaptive sample-anchor similarity learning and latent embeddings learning can obtain discriminative representation for clustering.

5.3 Parameter Analysis

In this subsection, we analyze the influence of model parameters α and β on clustering performance. Figure 2 shows the change in ACC values with different combination of α and β on BBCSport and VGGFace datasets. The horizontal axes are scaled using $\log_{10}$. We can see that the parameters influence the algorithm performance. Generally, when α and β are selected from $[10^2, 10^5]$ and $[10^0, 10^2]$, the results are not sensitive to the parameters and satisfactory performance is obtained. Moreover, the number of anchors is fixed as $k = 5c$ in all experiments. For a comprehensive evaluation, we also investigate the influence of k and display the results in Figure 2. In general, when m is selected from $[4, 6]$, the clustering results are desirable.

5.4 Convergence and Time Comparison

An alternate optimization algorithm is developed to solve the proposed model. Since each sub-problem is solved with a closed-form solution, the algorithm can converge to a local minimum finally. In this subsection, we experimentally illustrate the algorithm convergence characteristic by showing the change of objective function value. Figure 3 plots the convergence curves on BBCSport and VGGFace datasets. The curve of I-stage is obtained from the initial data chunk (round 1), while the curve of C-stage is derived from the subsequent data chunk (round 2). As can be observed, the objective function value declines swiftly on two datasets, allowing the algorithm to converge within only 10 iterations.

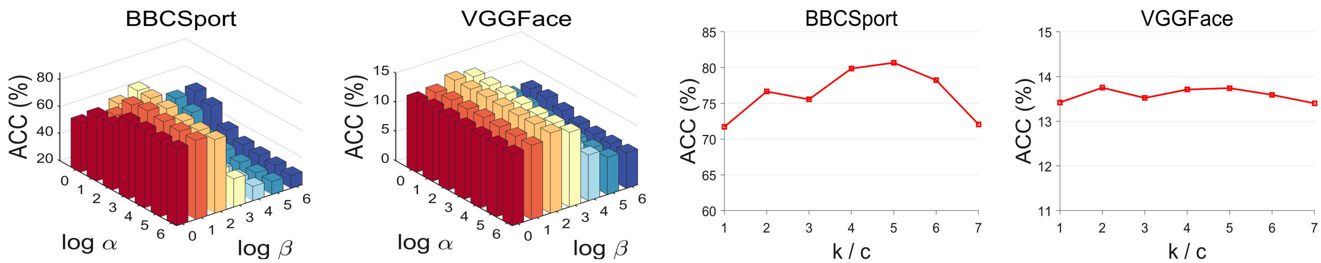

Figure 2: Parameter analysis of ACMVC w.r.t. α, β and k on BBCSport and VGGFace datasets.

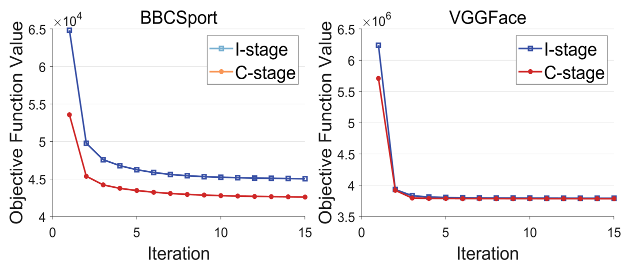

Figure 3: Convergence curves on BBCSport and VGGFace datasets.

Dataset	Ours-A	Ours-V	Ours-F	Ours
BBCSport	48.58(0.09)	37.39(0.94)	74.41(8.13)	80.68(0.10)
Mfeat	73.36(0.84)	51.89(1.92)	77.25(5.23)	86.18(0.92)
Wiki	53.41(0.00)	32.32(0.71)	54.12(1.17)	55.10(1.67)
MITIndoor	34.37(1.41)	12.21(0.50)	45.62(2.49)	46.68(1.25)
Caltech101	24.25(0.72)	8.72(0.35)	25.17(1.33)	26.46(0.95)
Fashion	73.08(1.32)	48.87(0.78)	75.73(0.07)	75.77(0.05)
VGGFace	11.78(0.32)	4.44(0.04)	13.71(0.57)	13.74(0.38)

Table 3: Ablation study results w.r.t. ACC (%) values.

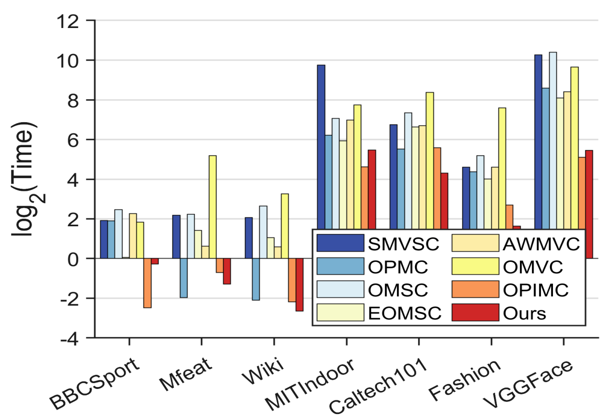

Figure 4: Running time comparison on seven datasets.

We also compare the running time (second) of different methods. SMVSC, OPMC, OMSC, EOMSC, and AWMVC belong to static MVC algorithm, and they have to undergo a complete re-learning process when new data arrives, while OMVC, OPIMC and our ACMVC are in an online learning manner. Figure 4 shows the running time comparison results, with the y-axis rescaled using $\log_2$. OMVC is not efficient because it adopts a Hessian matrix based projected gradient descent algorithm to solve the non-negative matrix factorization problem. OPIMC and our method stand out as the two fastest algorithms in terms of speed. For example, on VGGFace dataset, the running time of our method is less than 50 seconds. These results demonstrate the efficiency our proposed method.

5.5 Ablation Study

To efficiently handle streaming data, our ACMVC transfers the historical multi-level anchor information, i.e., $\{\mathbf{A}_t^v\}_{v=1}^m$ and $\mathbf{V}_t$. In this subsection, we conduct ablation study to evaluate the effectiveness of the two knowledge reuse regularization. In specific, based on Eq. (5), we establish two variants Ours-A and Ours-V. Ours-A excludes the $\sum_{v=1}^m \|\mathbf{A}_t^v - \mathbf{A}_{t-1}^v\|_F^2$ term, and Ours-V excludes the $\|\mathbf{V}_t - \mathbf{V}_{t-1}\|_F^2$ term. Besides, we also evaluate the performance of Eq. (3), which adopts fixed anchor information for knowledge transfer without regularization, and we denote it as Ours-F. Table 3 reports the clustering results w.r.t. ACC of the three variants and our original method on seven datasets. It should be noted that the model parameters of all variants are also tuned by grid search to report the best performance for a fair comparison. From Table 3, we can see that our proposed model achieves the best performance in all scenarios. Ours-V performs the worst, because the latent anchor embedding is strongly associated with sample embedding. Without the guidance of historical anchor embedding, the inter-chunk sample relationships cannot be captured. Ours-F gets competitive results because it also uses the historical multi-level anchor knowledge. Our method still outperforms Ours-F, proving the effectiveness of anchor consistency regularization.

6 Conclusion

In this paper, we propose a consistent anchor guided continual multi-view clustering method called ACMVC to deal with streaming multi-view data. ACMVC consists of two stages. In initial learning stage, an efficient low-rank anchor graph based model is constructed. In continual learning stage, the historical model is refined by adding consistency regularization to transfer the learned multi-level anchor information. Experiments demonstrate the superiority and efficiency of our method compared with state-of-the-art approaches, and ablation studies prove the effectiveness of our designed knowledge reuse mechanism. In the future, we will consider other advanced knowledge transfer strategies.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China under Grants 62176116, 62073160, 62276136, and the Natural Science Foundation of the Jiangsu Higher Education Institutions of China under Grant 20KJA520006.

References

[Cai *et al.*, 2022] Jinyu Cai, Jicong Fan, Wenzhong Guo, Shiping Wang, Yunhe Zhang, and Zhao Zhang. Efficient deep embedded subspace clustering. In *CVPR*, pages 21–30, 2022.

[Cai *et al.*, 2024] Jinyu Cai, Yunhe Zhang, Shiping Wang, Jicong Fan, and Wenzhong Guo. Wasserstein embedding learning for deep clustering: A generative approach. *IEEE TMM*, 2024.

[Chen *et al.*, 2022] Mansheng Chen, Chang-Dong Wang, Dong Huang, Jian-Huang Lai, and Philip S. Yu. Efficient orthogonal multi-view subspace clustering. In *ACM SIGKDD*, pages 127–135, 2022.

[Hou *et al.*, 2023] Chenping Hou, Shilin Gu, Chao Xu, and Yuhua Qian. Incremental learning for simultaneous augmentation of feature and class. *IEEE TPAMI*, 45(12):14789–14806, 2023.

[Hu and Chen, 2019] Menglei Hu and Songcan Chen. One-pass incomplete multi-view clustering. In *AAAI*, volume 33, pages 3838–3845, 2019.

[Huang *et al.*, 2019] Zhenyu Huang, Joey Tianyi Zhou, Xi Peng, Changqing Zhang, Hongyuan Zhu, and Jiancheng Lv. Multi-view spectral clustering network. In *IJCAI*, pages 2563–2569, 2019.

[Huang *et al.*, 2022] Shudong Huang, Hongjie Wu, Yazhou Ren, Ivor W. Tsang, Zenglin Xu, Wentao Feng, and Jiancheng Lv. Multi-view subspace clustering on topological manifold. In *NeurIPS*, 2022.

[Jiang *et al.*, 2023] Bingbing Jiang, Chenglong Zhang, Yan Zhong, Yi Liu, Yingwei Zhang, Xingyu Wu, and Weiguo Sheng. Adaptive collaborative fusion for multi-view semi-supervised classification. *Inf. Fusion*, 96:37–50, 2023.

[Jiang *et al.*, 2024] Bingbing Jiang, Xingyu Wu, Xiren Zhou, Yi Liu, Anthony G. Cohn, Weiguo Sheng, and Huanhuan Chen. Semi-supervised multiview feature selection with adaptive graph learning. *IEEE TNNLS*, 35(3):3615–3629, 2024.

[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 *AAAI*, pages 4412–4419, 2020.

[Lange *et al.*, 2022] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Gregory G. Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE TPAMI*, 44(7):3366–3385, 2022.

[Li and Gu, 2023] Diyang Li and Bin Gu. When online learning meets ODE: learning without forgetting on variable feature space. In *AAAI*, pages 8545–8553, 2023.

[Li and Hoiem, 2018] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE TPAMI*, 40(12):2935–2947, 2018.

[Li *et al.*, 2015] Yeqing Li, Feiping Nie, Heng Huang, and Junzhou Huang. Large-scale multi-view spectral clustering via bipartite graph. In *AAAI*, pages 2750–2756, 2015.

[Li *et al.*, 2022a] Xingfeng Li, Quansen Sun, Zhenwen Ren, and Yinghui Sun. Dynamic incomplete multi-view imputing and clustering. In *ACM MM*, pages 3412–3420. ACM, 2022.

[Li *et al.*, 2022b] Xuelong Li, Han Zhang, Rong Wang, and Feiping Nie. Multiview clustering: A scalable and parameter-free bipartite graph fusion method. *IEEE TPAMI*, 44(1):330–344, 2022.

[Li *et al.*, 2023a] Huaxiong Li, Chao Zhang, Xiuyi Jia, Yang Gao, and Chunlin Chen. Adaptive label correlation based asymmetric discrete hashing for cross-modal retrieval. *IEEE TKDE*, 35(2):1185–1199, 2023.

[Li *et al.*, 2023b] Xingfeng Li, Yinghui Sun, Quansen Sun, Zhenwen Ren, and Yuan Sun. Cross-view graph matching guided anchor alignment for incomplete multi-view clustering. *Inf. Fusion*, 100:101941, 2023.

[Lin *et al.*, 2021] Yijie Lin, Yuanbiao Gou, Zitao Liu, Boyun Li, Jiancheng Lv, and Xi Peng. COMPLETER: incomplete multi-view clustering via contrastive prediction. In *CVPR*, pages 11174–11183, 2021.

[Liu *et al.*, 2013] Guangcan Liu, Zhouchen Lin, Shuicheng Yan, Ju Sun, Yong Yu, and Yi Ma. Robust recovery of subspace structures by low-rank representation. *IEEE TPAMI*, 35(1):171–184, 2013.

[Liu *et al.*, 2021a] Jiyuan Liu, Xinwang Liu, Yuexiang Yang, Li Liu, Siqi Wang, Weixuan Liang, and Jiangyong Shi. One-pass multi-view clustering for large-scale data. In *ICCV*, pages 12324–12333, 2021.

[Liu *et al.*, 2021b] Xinwang Liu, Li Liu, Qing Liao, Siwei Wang, Yi Zhang, Wenxuan Tu, Chang Tang, Jiyuan Liu, and En Zhu. One pass late fusion multi-view clustering. In *ICML*, volume 139, pages 6850–6859, 2021.

[Liu *et al.*, 2022] Suyuan Liu, Siwei Wang, Pei Zhang, Kai Xu, Xinwang Liu, Changwang Zhang, and Feng Gao. Efficient one-pass multi-view subspace clustering with consensus anchors. In *AAAI*, pages 7576–7584, 2022.

[Qin *et al.*, 2024] Yalan Qin, Nan Pu, and Hanzhou Wu. Elastic multi-view subspace clustering with pairwise and high-order correlations. *IEEE TKDE*, 36(2):556–568, 2024.

[Ren *et al.*, 2021] Zhenwen Ren, Quansen Sun, and Dong Wei. Multiple kernel clustering with kernel k-means coupled graph tensor learning. In *AAAI*, pages 9411–9418, 2021.

[Shao *et al.*, 2016] Weixiang Shao, Lifang He, Chun-Ta Lu, and Philip S. Yu. Online multi-view clustering with incomplete views. In *ICBD*, pages 1012–1017, 2016.

[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 *ACM MM*, pages 3528–3536, 2021.

[Sun *et al.*, 2023] Yuan Sun, Dezhong Peng, Jian Dai, and Zhenwen Ren. Stepwise refinement short hashing for image retrieval. In *ACM MM*, pages 6501–6509, 2023.

[Sun *et al.*, 2024a] Yuan Sun, Jian Dai, Zhenwen Ren, Yingke Chen, Dezhong Peng, and Peng Hu. Dual self-paced cross-modal hashing. In *AAAI*, volume 38, pages 15184–15192, 2024.

[Sun *et al.*, 2024b] Yuan Sun, Zhenwen Ren, Peng Hu, Dezhong Peng, and Xu Wang. Hierarchical consensus hashing for cross-modal retrieval. *IEEE TMM*, 26:824–836, 2024.

[Tang *et al.*, 2019] Chang Tang, Xinzhong Zhu, Xinwang Liu, Miaomiao Li, Pichao Wang, Changqing Zhang, and Lizhe Wang. Learning a joint affinity graph for multiview subspace clustering. *IEEE TMM*, 21(7):1724–1736, 2019.

[Trosten *et al.*, 2021] Daniel J. Trosten, Sigurd Løkse, Robert Jenssen, and Michael Kampffmeyer. Reconsidering representation alignment for multi-view clustering. In *CVPR*, pages 1255–1265, 2021.

[Wan *et al.*, 2022] Xinhang Wan, Jiyuan Liu, Weixuan Liang, Xinwang Liu, Yi Wen, and En Zhu. Continual multi-view clustering. In *ACM MM*, pages 3676–3684, 2022.

[Wan *et al.*, 2023] Xinhang Wan, Xinwang Liu, Jiyuan Liu, Siwei Wang, Yi Wen, Weixuan Liang, En Zhu, Zhe Liu, and Lu Zhou. Auto-weighted multi-view clustering for large-scale data. In *AAAI*, pages 10078–10086, 2023.

[Wang *et al.*, 2020] Hao Wang, Yan Yang, and Bing Liu. GMC: graph-based multi-view clustering. *IEEE TKDE*, 32(6):1116–1129, 2020.

[Wang *et al.*, 2023] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *arXiv preprint arXiv:2302.00487*, 2023.

[Yin *et al.*, 2021] Hongwei Yin, Wenjun Hu, Zhao Zhang, Jungang Lou, and Minmin Miao. Incremental multi-view spectral clustering with sparse and connected graph learning. *Neural Networks*, 144:260–270, 2021.

[Zhang *et al.*, 2022a] Chao Zhang, Huaxiong Li, Caihua Chen, Xiuyi Jia, and Chunlin Chen. Low-rank tensor regularized views recovery for incomplete multiview clustering. *IEEE TNNLS*, 2022.

[Zhang *et al.*, 2022b] Chao Zhang, Huaxiong Li, Chunlin Chen, Yuhua Qian, and Xianzhong Zhou. Enhanced group sparse regularized nonconvex regression for face recognition. *IEEE TPAMI*, 44(5):2438–2452, 2022.

[Zhang *et al.*, 2023a] Chao Zhang, Huaxiong Li, Yang Gao, and Chunlin Chen. Weakly-supervised enhanced semantic-aware hashing for cross-modal retrieval. *IEEE TKDE*, 35(6):6475–6488, 2023.

[Zhang *et al.*, 2023b] Chao Zhang, Huaxiong Li, Wei Lv, Zizheng Huang, Yang Gao, and Chunlin Chen. Enhanced tensor low-rank and sparse representation recovery for incomplete multi-view clustering. In *AAAI*, pages 11174–11182, 2023.

[Zhang *et al.*, 2024] Chao Zhang, Xiuyi Jia, Zechao Li, Chunlin Chen, and Huaxiong Li. Learning cluster-wise anchors for multi-view clustering. In *AAAI*, pages 16696–16704, 2024.

[Zhou *et al.*, 2022] Da-Wei Zhou, Yang Yang, and De-Chuan Zhan. Learning to classify with incremental new class. *IEEE TNNLS*, 33(6):2429–2443, 2022.