Resisting Label Drift: Real-Time Multi-View Clustering with Semantic Consistency

Qi Liu[1] , **Suyuan Liu**[1] , **Hao Tan**[1] , **Yangfan Du**[1] , **Bowen Zhang**[1] , **Wenpeng Lu**[2] and **Xinwang Liu**[1*]

[1]National University of Defense Technology
[2]Qilu University of Technology (Shandong Academy of Sciences)
{liuqiget, tanhao_cs}@163.com, {suyuanliu, xinwangliu}@nudt.edu.cn, du1205340311@qq.com, zhangzbw@yeah.net, wenpeng.lu@qlu.edu.cn

Abstract

Real-time clustering of dynamic multi-view data streams is a critical yet challenging task in open-world applications. While several methods have been proposed to address this task, most of them extract features incrementally but fail to output instant clustering results for the current batch. In addition, they neglect semantic consistency, causing the cluster labels of identical concepts to drift unpredictably due to independent processing. To address these limitations, we propose a real-time semantic consistent incremental multi-view clustering framework. Specifically, we construct a compact historical knowledge base via an adaptive diversity-aware selection mechanism, which guides the clustering of incoming data, enabling immediate inference without accessing the full history. Furthermore, we introduce a semantic alignment strategy based on consensus centers to ensure robust label consistency over time. Extensive experiments demonstrate the effectiveness and efficiency of our proposed method.

1 Introduction

In practical applications, data is routinely gathered from various independent sources, resulting in multi-view data which inherently exhibits heterogeneity across different perspectives or views [Liang *et al.*, 2025; Liu *et al.*, 2025]. Due to the increasing prevalence of open-world problems involving such data, especially in the unsupervised domain, learning methodologies tailored for this data format have become a prominent area of research. Clustering, a foundational unsupervised task, aims to partition data into non-overlapping groups [Liu *et al.*, 2023; Xiao *et al.*, 2023]. In the context of multiple views, Multi-view clustering (MVC) approaches are designed to effectively utilize the data by uncovering both the consistency and complementarity that exist across the views. A wide array of MVC techniques has been developed, encompassing subspace-based methods [Wei *et al.*, 2023; Peng *et al.*, 2022; Qin *et al.*, 2025], kernel-based methods [Wu *et al.*, 2025; Zhang *et al.*, 2025b], graph-based methods [Pan and Kang, 2021; Xia *et al.*, 2022; He *et al.*, 2025], and deep learning-based architectures [Cui *et al.*, 2025; Wang *et al.*, 2025], among others. Despite their achievements, most of these fall under the category of static MVC, meaning they are restricted to scenarios where the dataset is fixed. This condition is frequently violated in real-world settings.

A more challenging yet common scenario involves data that is continuously accumulated and updated, forming a dynamic data stream [De Lange *et al.*, 2021]. For instance, social media platforms like YouTube continuously ingest multi-view streams (images, texts, videos) generated by users every second. Similarly, financial markets rely on high-frequency feeds of stock prices, news headlines, and trading volumes. Analyzing and clustering such continuously evolving streams presents significantly greater challenges compared to static scenarios.

A naive approach to address this challenge is to accumulate all observed data and repeatedly apply conventional MVC techniques to the aggregated dataset. However, this strategy is highly inefficient, incurring prohibitive storage costs and computational overhead. Moreover, it necessitates repeated access to historical data, which may be restricted due to privacy or storage constraints. To mitigate these issues, several incremental multi-view clustering algorithms have been proposed. Nevertheless, they suffer from two critical limitations. First, most methods lack real-time capability, as they fail to output instant clustering results for the current batch, often requiring offline post-processing. Second, and more critically, they overlook semantic consistency across data batches. For instance, a cluster representing the semantic concept of "Bird" might be assigned the label "ID-1" in the first batch, but the exact same object type could be randomly reassigned to "ID-5" in the next batch simply due to initialization differences. Without a mechanism to map cluster labels across time steps, these methods cannot consistently track evolving clusters, failing to provide interpretable, semantically continuous results.

To address these challenges, we propose a real-time semantic consistency incremental multi-view clustering (RTSC) framework. Our method integrates incoming data with a compact historical knowledge base via a consensus bipartite graph under a k-connectivity constraint, enabling real-time cluster-

*Corresponding author.

ing without the need for post-processing. Simultaneously, we employ an adaptive diversity-aware selection mechanism to dynamically update the knowledge base. Furthermore, we introduce a semantic alignment strategy based on consensus centers to ensure label consistency over time.

The following three points detail the primary contributions of this paper:

(1) We introduce an adaptive, diversity-aware sample selection mechanism that dynamically updates the knowledge base by balancing representativeness and redundancy to efficiently retain valuable historical structure.

(2) We develop a semantic alignment technique to ensure that cluster labels maintain consistent meaning over time. To our knowledge, this is the first work to explicitly address the semantic consistency of cluster labels in incremental multi-view clustering.

(3) We introduce a multi-view clustering framework for incremental instance scenarios. Experimental results across multiple datasets demonstrate the effectiveness and efficiency of the proposed framework.

2 Related Work

2.1 Static Multi-view Clustering

In recent years, diverse methodologies have emerged to address MVC, encompassing techniques grounded in matrix factorization, graph-based strategies, and deep learning frameworks. Methods utilizing matrix factorization typically leverage factorization techniques to derive a shared latent representation for clustering purposes. For instance, [Yang *et al.*, 2024] proposed an efficient and scalable anchor-based multi-view clustering method that directly obtains low-dimensional consensus embedding by factorizing the large-scale consensus graph. In a similar vein, [Liang *et al.*, 2020] introduced a non-negative matrix factorization model with co-orthogonal constraints for multi-view clustering, which enhances the orthogonality of the basis and representation matrices to improve clustering performance. Graph-based approaches construct individual graph embedding matrices for each view by leveraging intrinsic data structural information, followed by fusing them into a unified graph matrix through learned weights. For example, [Tan *et al.*, 2023] proposed a unified multi-view clustering framework that learns the manifold topological structure of data and explores sample-level cross-view consistency to better preserve consistent local structures across views. [Liu *et al.*, 2024] learned a consistent anchor graph with a k-connectivity constraint to directly generate cluster labels while filtering out view-specific noise. In recent years, following the advancements in deep learning, several studies have adopted deep neural networks to learn high-level feature representations for this task [Xu *et al.*, 2025], [Bian *et al.*, 2025], [Cui *et al.*, 2026]. While these varied methods have demonstrated promising results, most are tailored for static MVC configurations, operating under the assumption that all data instances and views are fully available in advance. This premise, however, is frequently unrealistic in open-world environments.

2.2 Dynamic Multi-view Clustering

In practical applications, data are often collected continuously, resulting in evolving data streams. Conventional static MVC approaches are generally ineffective in handling such dynamic data, as they typically require storing and reprocessing all available data whenever new samples arrive. In the instance incremental learning scenario, researchers conducted preliminary explorations. For instance, [Shao *et al.*, 2016] laid foundational work by adapting the conventional non-negative matrix factorization into an online framework by introducing a regularization term that enforces a consistent base matrix. Building on matrix factorization techniques, [Hu and Chen, 2019] subsequently leveraged matrix factorization to simultaneously learn cluster indicators and a unified base matrix, thereby facilitating effective knowledge transfer. Taking a different approach, [Huang *et al.*, 2019] tackled this problem by developing a multi-view support vector domain description model coupled with a multi-view cluster labeling model. Further diversifying the methodologies, [Zhang *et al.*, 2024] utilized multi-level semantic anchor information as prior knowledge to steer the representation learning of incoming data. Most recently, [Zhang *et al.*, 2025a] addressed a more complex challenge by employing reusable embedding and centroid modules with consistency regularization to efficiently handle the simultaneous arrival of new instances and views. Although the methods mentioned above have made some progress, they all perform clustering only after processing all the data. In practical applications, the subsequent data in the current batch is unpredictable, and it is often necessary to provide clustering results in real time after a batch of data arrives. Furthermore, they do not consider the semantic drift problem between clustering results of different batches of data.

3 The Proposed Framework

This section introduces a real-time semantic consistency incremental multi-view clustering framework designed for the incremental instance scenario in which samples arrive in sequence and cannot be fully revisited.

Notations	Descriptions
V, m, k	Amount of views, anchors and clusters
n^t	Amount of samples at time t
l	Size of the historical knowledge base
d_v	Dimension of the v-th view
d	Sum of view features
$\mathbf{X}_v^t \in \mathbb{R}^{d_v \times n^t}$	Newly arrived data at time t
$\mathbf{C}_v^t \in \mathbb{R}^{d_v \times m}$	Anchor points matrix at time t
$\mathbf{H}^t \in \mathbb{R}^{m \times (n^t + l)}$	The consensus bipartite graph matrix at time t
$\mathbf{K}_v^t \in \mathbb{R}^{d_v \times l}$	Knowledge base matrix of view v at time t

Table 1: Description of Symbols Used in the Article

.

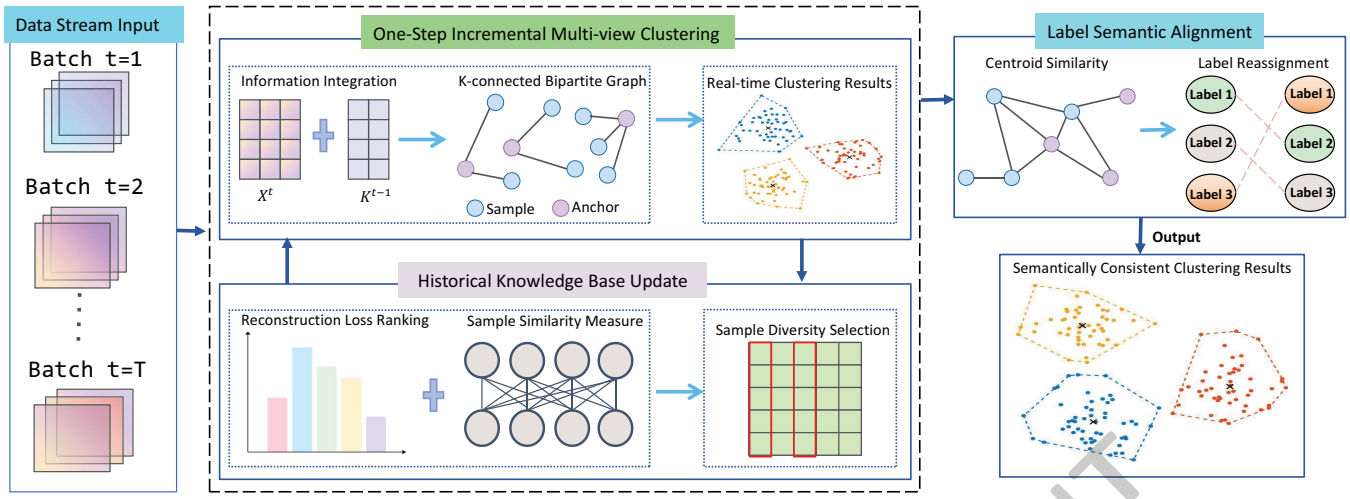


Figure 1: The proposed incremental multi-view clustering framework RTSC.

3.1 Notations and Problem Statement

Table 1 provides an overview of the main symbols employed in this article. This work focuses on multi-view clustering in the incremental instance scenario. In each iteration, a batch of multi-view observation samples arrives. The goal is to obtain k semantic labels through real-time clustering for each batch of data, with the clustering labels having the same semantic meaning across different batches. The main challenges in achieving this goal are as follows: 1) How to perform real-time clustering after a new batch of data arrives. 2) How to retain valuable historical information with limited computing resources. 3) How to achieve label semantic alignment in clustering across different batches. To address the challenges mentioned above, the proposed method designs three core components to tackle these problems, respectively.

3.2 One-step Incremental Multi-view Clustering with Memory Replay

Upon the arrival of the t-th batch data $\mathbf{X}^t = \{\mathbf{X}_v^t\}_{v=1}^V$, we integrate the historical knowledge base with the new data for joint clustering. The clustering process of newly arriving data is softly guided by utilizing the implicit clustering structure within the historical knowledge base. Simultaneously, the historical knowledge base is integrated with the newly arriving data for unified graph learning, in order to present a clear clustering structure and achieve real-time, one-step clustering. Based on the above ideas, we define the following optimization objectives:

$$\begin{aligned}
&\min_{\alpha_v, \mathbf{C}_v^t, \mathbf{H}^t} \sum_{v=1}^{V} \alpha_v^2 \|[\mathbf{X}_v^t, \mathbf{K}_v^{t-1}] - \mathbf{C}_v^t \mathbf{H}^t\|_F^2, \\
&\text{s.t. } \alpha^\top \mathbf{1} = 1,\ \alpha \geq 0, \mathbf{H}^t \geq 0, {\mathbf{H}^t}^\mathrm{T} \mathbf{1} = \mathbf{1}, \\
&\quad {\mathbf{C}_v^t}^\mathrm{T} \mathbf{C}_v^t = \mathbf{I}_m,\ \mathrm{rank}(L_S^t) = n^t + l + m - k, \qquad (1)
\end{aligned}$$

where the weight α is used to balance the effects of different views, $\mathbf{K}_v^{t-1}$ represents the historical knowledge base, and its construction method is described in section 3.3, $\mathbf{C}_v^t$ and $\mathbf{H}^t$ represent the anchors matrix and anchor graph matrix at time t, respectively. The augmented graph matrix $\mathbf{S}^t$ is defined as a block matrix consisting of four blocks. The two diagonal blocks are zero matrices, the upper-right block is the transpose of $\mathbf{H}^t$, and the lower-left block is $\mathbf{H}^t$. L_S^t is the normalized Laplace matrix of $\mathbf{S}^t$.

The number of connected components in $\mathbf{S}^t$ matches the number of zero eigenvalues of L_S^t. Since $\mathbf{H}^t$ is constructed from $\mathbf{S}^t$, both share identical connectivity. Consequently, enforcing a rank condition on L_S^t is essentially equivalent to ensuring a specific connectivity structure in $\mathbf{H}^t$. With $\mathbf{H}^t$ designed to contain exactly k connected components, the clustering assignments can be obtained directly, without any additional post-processing [Nie *et al.*, 2016]. Each component corresponds to a group of nodes that are mutually reachable, forming the real-time cluster partitions.

3.3 Historical Knowledge Base Information Selection

The constructed historical knowledge base should implicitly contain the real-time clustering structure of the data stream while maintaining the representativeness of historical information. Based on this, the historical knowledge base $\mathbf{K}_v^t$ should be adaptively updated with each new batch of data to adapt to the new data distribution. To enable the knowledge base to naturally embed the graph learning process, it should store a compact and easily learnable representative sample set. To ensure the knowledge base meets these characteristics, the specific process for constructing and updating the knowledge base information is designed as follows.

When $t = 1$, no prior information is available, and the historical knowledge base $\mathbf{K}_v^{t-1}$ is initialized as an empty set.

When $t > 1$, the process of building and updating the knowledge base is as follows. First, the current data $\mathbf{X}^t$ and the existing historical knowledge base $\mathbf{K}^{t-1}$ are merged to form unified set $\mathbf{R}^t = [\mathbf{X}^t, \mathbf{K}^{t-1}]$. Then, each sample x_i in $\mathbf{R}^t$ is evaluated using the reconstruction loss: $\mathrm{loss}(x_i) =$

$\sum_{v=1}^{V} \alpha_v \|\mathbf{R}^t_{v(:,i)} - \mathbf{C}^t_v \mathbf{H}^t_{:,i}\|_2^2$, where $\mathbf{C}^t_v$ and $\mathbf{H}^t_{:,i}$ are learned from objective function 1 at time t. A smaller reconstruction loss indicates that the sample is more easily learned as an anchor representation, thus carrying more stable and explicit structural information. The samples are ranked according to the magnitude of the reconstruction loss.

Simply selecting samples based on the magnitude of the reconstruction loss can lead to significant redundancy and class imbalance, as samples with smaller reconstruction losses are often concentrated in a few narrow regions of the feature space. To overcome this problem, we employ a diversity-aware selection mechanism. In each iteration, the sample with lower reconstruction loss is prioritized as a candidate sample. If its similarity to any selected sample exceeds a given threshold θ, it is discarded; otherwise, it is accepted and added to the updated knowledge base. This process is repeated iteratively until no other samples can be selected. The similarity threshold θ plays a crucial role in balancing diversity and representativeness. To determine an appropriate value corresponding to a given sampling rate, we employ a binary search strategy to automatically adjust θ. The detailed algorithm for building and updating the knowledge base is shown in Algorithm 1.

Algorithm 1 Knowledge base construction and update algorithm

Input: Multi-view data $\mathbf{X}^t$ at time t, anchors $\mathbf{C}^t = \{\mathbf{C}^t_v\}_{v=1}^V$ and anchor graph $\mathbf{H}^t$ learned at time t, historical knowledge base $\mathbf{K}^{t-1}$, knowledge base size l.
Output:Updated knowledge base $\mathbf{K}^t$ at time t.

1: Initialize the knowledge base as an empty set;
2: Add anchors $\mathbf{C}^t$ learned at time t to the new knowledge base $\mathbf{K}^t = [\mathbf{C}^t]$;
3: Calculate the reconstruction loss for each sample in $\mathbf{X}^t$ and $\mathbf{K}^{t-1}$ based on $\mathbf{C}^t$ and $\mathbf{H}^t$ and sort them;
4: Calculate the similarity between $\mathbf{X}^t$, $\mathbf{K}^{t-1}$, and $\mathbf{C}^t$;
5: Adaptively adjust similarity threshold θ using a binary-search strategy until knowledge base reaches size l;
6: Add samples in $\mathbf{X}^t$ and $\mathbf{K}^{t-1}$ to update knowledge base $\mathbf{K}^t$ in sequence based on their ranking and θ.

3.4 Label Semantic Alignment

To address the issue of semantic drift and ensure that cluster labels across successive data batches maintain a consistent meaning, we introduce a semantic alignment mechanism based on consensus cluster center matching. The core idea is to establish an optimal one-to-one mapping between the clusters identified in the previous batch and those in the current batch.

First, for the current batch $\mathbf{X}^t$ and its initial clustering result $\mathbf{Y}^t$, we compute the cluster centers for each view by taking the mean of samples belonging to each cluster. These view-specific centers are then concatenated vertically to form a unified multi-view center matrix $\mathbf{Z}^t_{\text{combined}} \in \mathbb{R}^{\sum d_v \times k}$.

Then, we use the consensus aligned multi-view centers $\mathbf{Z}_{\text{consensus}}$ and the newly computed centers $\mathbf{Z}^t_{\text{combined}}$ to determine the optimal correspondence. Specifically, this paper uses the cluster centers of the first batch of data as the consensus cluster centers. The gaussian kernel [Hu *et al.*, 2010] is applied to compute the similarity matrix $\mathbf{A} \in \mathbb{R}^{k \times k}$ between all pairs of consensus centers and current centers. This similarity matrix measures how closely the i-th consensus cluster center semantically matches the j-th current cluster center.

Then, the alignment problem is then converted into a maximum weight bipartite matching problem, which is efficiently solved using the Hungarian Algorithm on the cost matrix $\mathbf{M} = -\mathbf{A}$. The resulting optimal pairing yields a label mapping $\mathcal{M} : \{1, \ldots, k\} \to \{1, \ldots, k\}$.

Finally, the initial cluster labels Y^t are globally remapped using the derived mapping $\mathcal{M}$ to generate the semantically aligned labels $\mathbf{Y}^t_{\text{aligned}}$. This procedure ensures that when a cluster from the previous batch corresponds to a cluster in the current batch, all samples currently assigned to that cluster are consistently reassigned the label of the matching cluster from the previous batch.

4 Optimization Strategy

To solve the above optimization problem, we propose an alternative iterative algorithm that updates one variable while keeping the other variables fixed and thus updates all variables alternately.

4.1 Solve $\mathbf{C}^t_v$ by Fixing Other Variables

Let $\mathbf{R}^t_v = [\mathbf{X}^t_v, \mathbf{K}^{t-1}_v]$. To solve for $\mathbf{C}^t_v$ while keeping all other variables fixed, we consider the following subproblem:

$$\min_{\mathbf{C}^t_v} \sum_{v=1}^{V} \alpha_v^2 \|\mathbf{R}^t_v - \mathbf{C}^t_v \mathbf{H}^t\|_F^2, \quad \text{s.t. } \mathbf{C}^{t^{\mathrm{T}}}_v \mathbf{C}^t_v = \mathbf{I}_m. \tag{2}$$

Rewriting the Frobenius norm in Eq. (2) into the form of trace yields the following equivalent formula about $\mathbf{C}^t_v$:

$$\min_{\mathbf{C}^t_v} -2 \operatorname{Tr}\left(\mathbf{C}^{t^T}_v \mathbf{R}^t_v \mathbf{H}^{t^T}\right), \quad \text{s.t. } \mathbf{C}^{t^{\mathrm{T}}}_v \mathbf{C}^t_v = \mathbf{I}_m. \tag{3}$$

The above formula can be transformed into the following equivalent problem:

$$\max_{\mathbf{C}^t_v} \operatorname{Tr}\left(\mathbf{C}^{t^T}_v \mathbf{B}^t_v\right), \quad \text{s.t. } \mathbf{C}^{t^{\mathrm{T}}}_v \mathbf{C}^t_v = \mathbf{I}_m, \tag{4}$$

where $\mathbf{B}^t_v = \mathbf{R}^t_v \mathbf{H}^{t^T}$. Assuming that the singular value decomposition of $\mathbf{B}^t_v$ is given by $\mathbf{B}^t_v = \mathbf{U}_v \mathbf{\Sigma} \mathbf{V}^{\mathrm{T}}_v$, the optimal solution of $\mathbf{C}^t_v$ can then be derived as $\mathbf{C}^t_v = \mathbf{U}_v \mathbf{V}^{\mathrm{T}}_v$.

4.2 Solve $\mathbf{H}^t$ by Fixing Other Variables

When all other variables are held constant, finding $\mathbf{H}^t$ becomes equivalent to solving the following problem:

$$\min_{\mathbf{H}^t} \sum_{v=1}^{V} \alpha_v^2 \left(-2 \operatorname{Tr}\left(\mathbf{H}^{t^{\mathrm{T}}} \mathbf{C}^{t^{\mathrm{T}}}_v \mathbf{R}^t_v\right) + \operatorname{Tr}\left(\mathbf{H}^{t^{\mathrm{T}}} \mathbf{H}^t\right)\right),$$
$$\text{s.t.} \mathbf{H}^t \geq 0,\ \mathbf{H}^{t^{\mathrm{T}}} \mathbf{1} = \mathbf{1}, \operatorname{rank}(\mathbf{L}^t_S) = n^t + l + m - k. \tag{5}$$

By choosing a sufficiently large value for ϕ, Eq. (5) can be reformulated as the following optimization problem [Fan, 1949]:

$$\begin{aligned}&\min_{\mathbf{F}^t,\mathbf{H}^t}\sum_{v=1}^{V}\alpha_v^2\left(-2\,\mathrm{Tr}\left(\mathbf{H}^{t^{\mathrm{T}}}\mathbf{C}_v^{t^{\mathrm{T}}}\mathbf{R}_v^t\right)+\mathrm{Tr}\left(\mathbf{H}^{t^{\mathrm{T}}}\mathbf{H}^t\right)\right)\\&+\phi\mathrm{Tr}(\mathbf{F}^{t^{\mathrm{T}}}\mathbf{L}_S^t\mathbf{F}^t),\\&\text{s.t.}\mathbf{H}^t\geq 0,\ \mathbf{H}^{t^{\mathrm{T}}}\mathbf{1}=\mathbf{1},\mathbf{F}^{t^{\mathrm{T}}}\mathbf{F}^t=\mathbf{I}_k,\end{aligned}\tag{6}$$

where $\mathbf{F}^t\in\mathbb{R}^{(n^t+m+l)\times k}$ denotes the indicator matrix. Following reference [Liu *et al.*, 2022], we alternately optimize $\mathbf{F}^t$ and $\mathbf{H}^t$. After obtaining the optimal $\mathbf{F}^t$, we proceed to update $\mathbf{H}^t$ in Eq. (6). Following reference [Liu *et al.*, 2022], we optimize each column of $\mathbf{H}^t$ separately, transforming optimization Eq. (5) into the following form:

$$\min_{\mathbf{H}_{:,i}^{t^{\mathrm{T}}}\mathbf{1}=1,\mathbf{H}_{:,i}^t\geq 0}\left\|\mathbf{H}_{:,i}^{t^{\mathrm{T}}}-(\mathbf{G}_{i,:}^t-\frac{\phi}{2}\mathbf{b}_i)/\sum_{v=1}^{V}\alpha_v^2\right\|_2^2,\tag{7}$$

where $\mathbf{G}^t=\sum_{v=1}^{V}\alpha^2\mathbf{R}_v^{t^{\mathrm{T}}}\mathbf{C}_v^t$, $\mathbf{b}_i\in\mathbb{R}^m$ is a column vector, and $b_{ij}=\left\|\frac{\mathbf{f}_i^{(n^t+l)}}{\sqrt{\mathbf{D}^{(n^t+l)}(i,i)}}-\frac{\mathbf{f}_j^{(m)}}{\sqrt{\mathbf{D}^{(m)}(j,j)}}\right\|_2^2$ refers to its j-th element. Reference [Nie *et al.*, 2014] shows that Eq. (7) can be solved in closed form. More detailed optimization procedures for $\mathbf{C}_v^t$ are provided in the **Supplementary materials**.

Algorithm 2 Algorithm for RTSC

Input Multi-view data stream $\{\mathbf{X}^1,\mathbf{X}^2,\dots\mathbf{X}^T\}$; the number of clusters k.
Output Real-time clustering results with semantic alignment.

for $t=1:T$ **do**
  Initialize $\mathbf{C}_v^t$, α and $\mathbf{H}^t$;
  repeat
    Update $\mathbf{C}_v$ by solving Eq. (2);
    Update $\mathbf{H}^t$ by solving Eq. (7);
    Update α by via Eq. (9);
  until convergence
  Output the cluster label of $\mathbf{X}^t$ partitioned by $\mathbf{H}^t$;
  Update historical knowledge base $\mathbf{K}^t$ at time t according to Algorithm 1;
  Semantic alignment of cluster label according to clustering centers alignment.
end for

4.3 Solve α by Fixing Other Variables

When all other variables are held constant, finding α becomes equivalent to solving the following problem:

$$\min_{\alpha}\sum_{v=1}^{V}\alpha_v^2\mathbf{P}_v^t,\quad s.t.\alpha^T1=1,\alpha\geq 0,\tag{8}$$

where $\mathbf{P}_v^t=||\mathbf{R}_v^t-\mathbf{C}_v^t\mathbf{H}^t||_F^2$. Based on the Cauchy-Schwarz inequality, the value of α can be derived directly as

$$\alpha_v=\frac{1/\mathbf{P}_v^t}{\sum_{v=1}^{V}1/\mathbf{P}_v^t}.\tag{9}$$

4.4 Complexity Analysis

The calculation process of the proposed method is shown in Algorithm 2. The time complexity of the algorithm is mainly composed of the following aspects. The time complexity of updating $\mathbf{C}_v^t$ is $O(mn^td_v)$. The complexity of updating $\mathbf{H}^t$ is $O(m^2n^t+m^3+n^tmd)$. The time complexity of updating α is $O(1)$. The time complexity of updating the historical knowledge base $\mathbf{K}^t$ is $O((n^t)^2d)$. The time complexity of label semantic alignment is $O(k^2d+k^3+n^td)$. In general, the overall time complexity of the proposed algorithm is $O(\sum_{t=1}^{T}(n^t)^2d+m^2n+k^2d+mnd)$, where $n=\sum_{t=1}^{T}n^t$.

5 Experimental Analysis

In this section, we present a comprehensive set of experiments to evaluate the effectiveness of our proposed approach. Due to space constraints, we have placed the convergence and complexity analysis and parameters analysis in the **Supplementary materials**.

5.1 Datasets

To ensure a thorough evaluation, our experiments utilize six widely recognized multi-view benchmark datasets: Animal[1], NUSWIDEOBJ [Chua *et al.*, 2009], YoutubeFace_sel, YTF10[2], YTF20[3], and YTF50[4]. Detailed statistics of these real-world datasets are summarized in Table 2. To construct a streaming environment, the entire dataset is partitioned into multiple batches, with one batch being processed in each iteration. For each dataset, the number of data blocks is set to the number of samples divided by 2000 and rounded up.

Datasets	Rename	Samples	Features	Views	Classes
Animal	D1	11673	2689/2000/2001/2000	4	20
NUSWIDEOBJ	D2	30000	65/226/145/74/129	5	31
YoutubeFace_sel	D3	101499	64/512/64/647/838	5	31
YTF10	D4	38654	944/576/512/640	4	10
YTF20	D5	63896	944/576/512/640	4	20
YTF50	D6	126054	944/576/512/640	4	50

Table 2: Description of Datasets.

5.2 Experimental Design

We evaluated our proposed method by comparing it with seven other multi-view clustering methods proposed in recent years. These include three dynamic multi-view clustering algorithms: MVC-IIV [Zhang *et al.*, 2025a], ACMVC [Zhang *et al.*, 2024], and OMVC [Shao *et al.*, 2016], and four static multi-view clustering algorithms: FMGM [Lu *et al.*, 2024], NpGC [Yu *et al.*, 2024], LMVSC [Kang *et al.*, 2020], and FPMVS [Wang *et al.*, 2022].

Our method does not require adjusting coefficients to balance the effects of individual components. However, the

[1]https://github.com/wangsiwei2010/large scale multi-view clustering datasets.
[2]https://www.micc.unifi.it/resources/datasets/e-ytf/
[3]https://www.micc.unifi.it/resources/datasets/e-ytf/
[4]https://www.micc.unifi.it/resources/datasets/e-ytf/

Data	Metric	**Proposed**	MVC-IIV	ACMVC	OMVC	FMCM	NpGC	LMVSC	FPMVS
D1	ACC	**21.16 ± 0.36**	20.22 ± 0.16	18.90 ± 0.42	16.78 ± 0.96	16.69 ± 1.37	18.13 ± 0.76	17.50 ± 0.95	18.18 ± 0.78
	PUR	22.70 ± 0.45	23.77 ± 0.19	22.57 ± 0.48	19.48 ± 1.20	18.02 ± 1.17	21.69 ± 0.59	20.50 ± 1.01	20.47 ± 0.59
	Fscore	**14.87 ± 0.46**	12.43 ± 0.24	10.95 ± 0.23	11.61 ± 0.43	12.81 ± 0.47	10.61 ± 0.27	10.23 ± 0.38	13.29 ± 0.80
	ARI	**8.32 ± 0.59**	7.32 ± 0.26	5.78 ± 0.27	3.80 ± 0.84	4.16 ± 0.66	5.47 ± 0.28	4.67 ± 0.41	7.19 ± 0.85
D2	ACC	**20.22 ± 0.78**	17.71 ± 0.49	15.92 ± 0.65	15.11 ± 0.45	16.63 ± 1.13	14.77 ± 0.59	16.41 ± 0.58	18.78 ± 0.94
	PUR	25.85 ± 0.90	28.45 ± 0.70	24.48 ± 0.46	25.18 ± 0.71	21.86 ± 0.42	25.01 ± 0.67	25.37 ± 0.61	24.33 ± 0.87
	Fscore	**13.56 ± 0.52**	10.59 ± 0.50	9.07 ± 0.42	11.18 ± 0.07	11.28 ± 0.79	8.22 ± 0.26	9.69 ± 0.51	11.90 ± 0.56
	ARI	**6.97 ± 0.40**	6.39 ± 0.51	4.59 ± 0.42	3.92 ± 0.40	2.63 ± 0.63	4.11 ± 0.27	4.91 ± 0.50	5.25 ± 0.47
D3	ACC	**26.93 ± 0.20**	25.44 ± 0.93	22.49 ± 0.69	26.66 ± 0.28	13.45 ± 2.85	27.20 ± 1.11	17.73 ± 0.95	21.58 ± 1.28
	PUR	27.31 ± 0.18	34.41 ± 0.83	32.38 ± 0.65	30.90 ± 0.68	27.48 ± 0.49	35.61 ± 0.96	29.67 ± 0.73	30.84 ± 1.01
	Fscore	**16.74 ± 0.30**	10.47 ± 0.37	10.21 ± 0.37	16.31 ± 0.23	9.04 ± 1.82	10.82 ± 0.45	8.91 ± 0.39	12.71 ± 0.51
	ARI	1.63 ± 0.29	5.54 ± 0.39	4.81 ± 0.31	2.95 ± 0.37	0.89 ± 0.48	6.18 ± 0.48	2.48 ± 0.29	4.76 ± 0.56
D4	ACC	**81.80 ± 4.00**	77.52 ± 2.16	77.27 ± 1.51	64.74 ± 4.13	67.67 ± 5.13	73.77 ± 2.95	78.33 ± 3.43	73.86 ± 4.74
	PUR	**91.80 ± 4.73**	82.39 ± 1.44	80.86 ± 1.16	69.66 ± 3.59	72.04 ± 4.38	79.36 ± 2.47	81.18 ± 2.26	79.61 ± 4.31
	Fscore	**78.70 ± 4.41**	74.72 ± 1.86	74.18 ± 1.45	56.10 ± 5.91	63.16 ± 4.84	70.72 ± 2.96	72.42 ± 2.61	72.52 ± 4.82
	ARI	**76.27 ± 5.04**	71.54 ± 2.11	70.92 ± 1.65	50.02 ± 7.03	58.16 ± 5.77	67.00 ± 3.34	69.00 ± 2.98	69.09 ± 5.48
D5	ACC	**73.42 ± 1.90**	71.10 ± 0.76	69.59 ± 0.80	62.45 ± 3.56	61.99 ± 3.98	67.74 ± 1.70	68.77 ± 3.18	69.57 ± 3.52
	PUR	**84.97 ± 4.35**	76.33 ± 0.64	75.02 ± 0.70	68.39 ± 2.62	67.39 ± 3.54	74.24 ± 1.32	73.91 ± 2.14	74.08 ± 2.91
	Fscore	**70.56 ± 2.66**	65.70 ± 0.92	64.32 ± 0.88	50.51 ± 3.78	56.75 ± 4.34	62.49 ± 1.60	61.12 ± 2.47	65.80 ± 2.16
	ARI	**68.87 ± 2.88**	63.61 ± 0.99	62.12 ± 0.95	47.28 ± 4.05	53.89 ± 4.75	60.18 ± 1.73	58.72 ± 2.65	63.74 ± 2.32
D6	ACC	**71.68 ± 2.07**	70.38 ± 0.70	68.49 ± 0.65	63.77 ± 2.55	66.43 ± 2.39	66.44 ± 1.12	63.30 ± 2.48	67.51 ± 4.22
	PUR	**86.82 ± 3.48**	75.57 ± 0.62	74.46 ± 0.59	70.14 ± 1.97	70.63 ± 2.23	73.30 ± 0.85	69.51 ± 1.83	71.37 ± 4.18
	Fscore	**67.68 ± 4.11**	63.83 ± 0.85	61.51 ± 0.83	54.27 ± 3.78	59.76 ± 2.45	59.43 ± 1.26	48.65 ± 3.32	60.41 ± 5.98
	ARI	**66.98 ± 4.25**	62.96 ± 0.87	60.57 ± 0.86	53.16 ± 3.90	58.76 ± 2.54	58.44 ± 1.29	47.27 ± 3.44	59.42 ± 6.17

Table 3: Batch average clustering performance comparison (%) using different methods on six datasets.

number of anchors and the size of the historical knowledge base must be specified. In our experiments, the number of anchors is selected from the set $[k, 2k, 3k]$, and the size of the historical knowledge base is selected from the set $[m, n/4, n/2, 3n/4, n]$. The comparison method uses the same number of anchor points as the proposed method. The comparison methods adjust and select their optimal parameters for fairness. To assess clustering performance, we adopt four widely used metrics: Accuracy (ACC), Purity (PUR), F-score and Adjusted Rand Index (ARI).

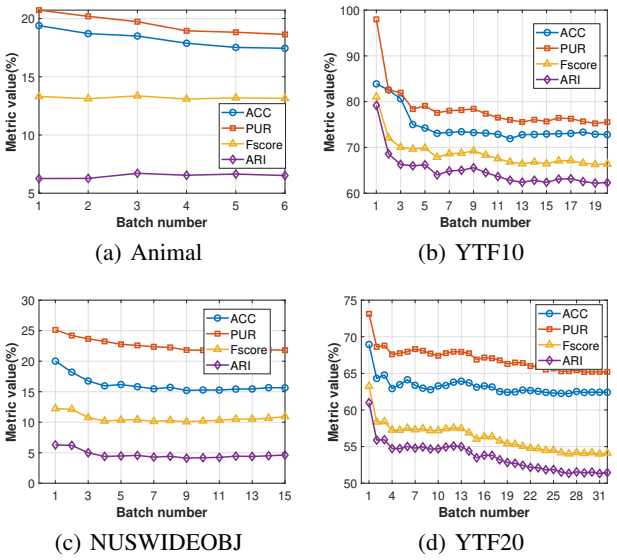
5.3 Clustering Results with Batch Average

Table 3 shows the clustering performance of different clustering algorithms on six multi-view datasets. We performed clustering on each batch of data separately and then averaged the clustering results across all batches.

The experimental evaluation demonstrates that the proposed method consistently outperforms baseline approaches across diverse datasets. Our method achieves the highest accuracy on all six datasets, demonstrating its clustering performance in streaming data environments. For example, on the YTF10 dataset, the framework achieves the highest ACC, marking a significant improvement of over 10% compared to algorithms such as OMVC and FMCM. This superiority extends to the Animal datasets, where the method records the top ACC, F-score and ARI across seven different comparison methods, as well as the NUSWIDEOBJ dataset. Furthermore, the proposed method outperforms the comparative methods on all clustering accuracy metrics across the YTF10, YTF20, and YTF50 datasets. The results presented in the table indicate that the incremental multi-view clustering algorithm generally surpasses its static counterpart, thereby validating the efficacy of methods specifically engineered for streaming data. Furthermore, the consistent performance observed across diverse datasets, including object and animal imagery as well as video-based facial data, highlights the broad generalizability of our framework. The integration of adaptive knowledge base updating enables the model to retain essential historical information without being overwhelmed by obsolete patterns. The consistent gains over strong baselines, including state-of-the-art incremental and non-incremental multi-view clustering methods, confirm that our proposed framework can effectively handle incremental multi-view scenarios and retain valuable historical information.

5.4 Overall Clustering Results with Semantic Alignment

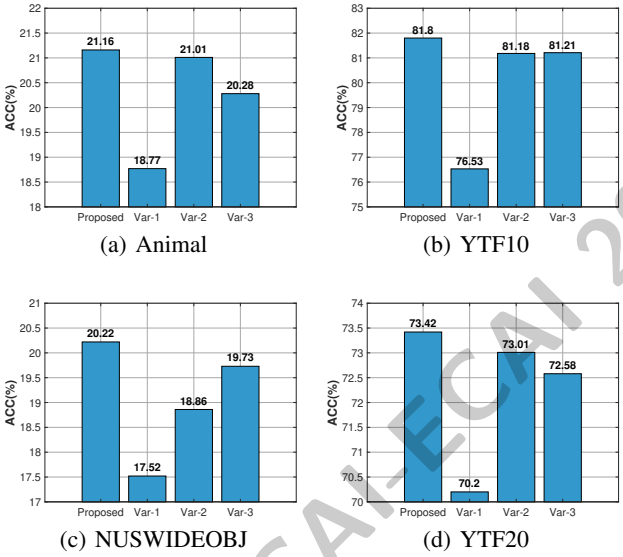
To evaluate the effectiveness of the proposed semantic alignment mechanism in mitigating label semantic drift across data batches, we conducted a semantic alignment experiment. For each batch of data, we obtain real-time cluster labels and align them with the previous labels. Once all batches of data have arrived, we combine all the real-time cluster labels from all batches to perform an overall clustering performance evaluation. For the comparison method, they are also implemented using the same real-time semantic alignment approach as proposed in this paper. We recorded the overall semantic alignment clustering results after all batches of data arrived, as shown in Table 4. As can be seen from the table, our method achieves the best clustering performance compared to other methods on most datasets and most metrics.



(a) Animal (b) YTF10

(c) NUSWIDEOBJ (d) YTF20

Figure 2: Performance metric changes after semantic alignment across batches.



(a) Animal (b) YTF10

(c) NUSWIDEOBJ (d) YTF20

Figure 3: Comparing the ACC of RTSC and its variants.

Furthermore, we demonstrate how the aligned clustering accuracy changes for four metrics with data arrives sequentially in batches, as shown in Figure 2. Due to space limitations, we present the semantic alignment results of the proposed method on four datasets: Animal, NUSWIDEOBJ, YTF10 and YTF20. As can be seen from the figure, the clustering accuracy after semantic alignment inevitably decreases slightly in the initial stage. However, as the number of data batches gradually increases, the accuracy tends to level off or even slightly improve, rather than decreasing monotonically. This demonstrates the effectiveness of the proposed semantic alignment method.

Data	Metric	**Proposed**	MVC-IIV	ACMVC	OMVC	FMCM	NpGC	LMVSC	FPMVS
D1	ACC	**17.45**	16.77	17.04	14.09	12.77	14.15	14.08	15.18
	PUR	18.64	**20.79**	20.12	15.96	15.50	17.80	17.55	17.59
	Fscore	**13.14**	11.04	10.23	10.82	8.89	8.99	8.52	12.51
	ARI	**6.52**	5.92	4.95	3.37	3.36	3.77	3.11	6.49
D2	ACC	**15.64**	13.72	12.05	12.03	9.86	11.29	12.96	13.53
	PUR	21.80	**23.83**	21.55	20.32	19.02	20.74	21.07	21.18
	Fscore	**10.91**	8.54	7.64	10.82	6.43	6.91	7.91	9.13
	ARI	**4.63**	4.40	3.45	2.04	2.19	2.79	3.69	3.46
D3	ACC	17.99	19.76	18.17	**20.33**	8.41	21.65	14.17	16.40
	PUR	26.64	31.17	29.98	29.80	27.16	**31.63**	28.72	27.95
	Fscore	**15.02**	8.80	8.66	14.97	5.98	8.93	6.64	10.58
	ARI	0.80	3.97	3.70	2.79	0.64	**4.30**	1.68	3.44
D4	ACC	**72.81**	69.83	71.50	58.06	53.01	65.08	67.71	66.19
	PUR	**75.52**	73.84	74.81	59.05	58.80	68.16	70.04	69.10
	Fscore	**66.37**	61.42	66.14	46.67	43.25	56.70	60.39	61.24
	ARI	62.31	56.79	**62.85**	40.05	36.50	51.56	55.64	56.56
D5	ACC	**62.42**	62.23	61.15	54.10	58.07	57.06	57.41	62.20
	PUR	65.22	**66.34**	64.76	56.44	60.93	63.48	61.32	65.01
	Fscore	**54.11**	51.61	49.72	38.34	48.95	45.02	46.90	52.60
	ARI	**51.46**	48.82	46.79	34.48	45.93	41.92	43.73	49.86
D6	ACC	**60.88**	60.71	57.97	51.73	57.09	55.49	48.37	56.04
	PUR	63.11	**64.14**	62.13	56.89	58.87	58.28	53.06	59.18
	Fscore	**49.42**	49.36	46.06	39.86	46.81	41.21	31.31	43.42
	ARI	**48.36**	48.30	44.84	38.49	45.57	39.90	29.69	42.13

Table 4: Comparison of overall clustering performance after semantic alignment (%) on six datasets.

5.5 Ablation Study

To evaluate the effectiveness of the proposed method, we conducted ablation experiments to assess the performance of the proposed historical knowledge base construction and update method. Specifically, we designed three variant methods to simplify the construction of the knowledge base. Var-1 eliminates the historical knowledge base construction process and uses only the current batch of data for clustering. Var-2 randomly selects historical samples as the historical knowledge base. Var-3 incorporates only historical sample information without adding learned anchor or centroid information when constructing the historical knowledge base.

The performance of the proposed method and its variants is evaluated using the ACC metric on the Animal, NUSWIDEOBJ, YTF10 and YTF20 datasets. As illustrated in Figure 3, our proposed method consistently achieves the highest accuracy, outperforming all three variants on every dataset. These results confirm that each proposed component of the historical knowledge base construction and update is essential to the overall effectiveness of the framework.

6 Conclusion

In this paper, we proposed a novel a real-time semantic consistency incremental multi-view clustering framework to address the challenges of efficiency and label inconsistency in dynamic data streams. By employing a consensus bipartite graph with a k-connectivity constraint, our method enables real-time clustering without additional post-processing. Key to our approach is an adaptive diversity-aware selection mechanism that maintains a compact and representative knowledge base, alongside a semantic alignment strategy that prevents label drift across sequential batches. Future research will explore adapting this framework to handle data streams with an evolving number of clusters and enhancing its robustness against missing views.

Acknowledgments

This work is supported by the National Science Fund for Distinguished Young Scholars of China (No. 62325604) and the National Natural Science Foundation of China (No. 62276271, 62441618, 625B2182).

References

[Bian *et al.*, 2025] Jintang Bian, Yixiang Lin, Xiaohua Xie, Chang-Dong Wang, Lingxiao Yang, Jian-Huang Lai, and Feiping Nie. Multilevel contrastive multiview clustering with dual self-supervised learning. *IEEE Transactions on Neural Networks and Learning Systems*, 36(6):10422–10436, 2025.

[Chua *et al.*, 2009] Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yantao Zheng. Nus-wide: A real-world web image database from national university of singapore. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, pages 1–9, 2009.

[Cui *et al.*, 2025] Jinrong Cui, Bang Liufu, Chongjie Dong, Jingcheng Ke, and Jie Wen. Deep multi-view clustering with meta information compression. *IEEE Transactions on Image Processing*, 34:7514–7527, 2025.

[Cui *et al.*, 2026] Jinrong Cui, Xiaohuang Wu, Haitao Zhang, Chongjie Dong, and Jie Wen. Structure-guided deep multi-view clustering. *Information Fusion*, 125:103461, 2026.

[De Lange *et al.*, 2021] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3366–3385, 2021.

[Fan, 1949] Ky Fan. On a theorem of weyl concerning eigenvalues of linear transformations. *Proceedings of the National Academy of Sciences*, 35(11):652–655, 1949.

[He *et al.*, 2025] Hongqing He, Jie Xu, Guoqiu Wen, Yazhou Ren, Na Zhao, and Xiaofeng Zhu. Graph embedded contrastive learning for multi-view clustering. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pages 5336–5344, 2025.

[Hu and Chen, 2019] Menglei Hu and Songcan Chen. One-pass incomplete multi-view clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3838–3845, 2019.

[Hu *et al.*, 2010] Qinghua Hu, Lei Zhang, Degang Chen, Witold Pedrycz, and Daren Yu. Gaussian kernel based fuzzy rough sets: model, uncertainty measures and applications. *International Journal of Approximate Reasoning*, 51(4):453–471, 2010.

[Huang *et al.*, 2019] Ling Huang, Chang-Dong Wang, Hong-Yang Chao, and Philip S Yu. MVStream: Multi-view data stream clustering. *IEEE Transactions on Neural Networks and Learning Systems*, 31(9):3482–3496, 2019.

[Kang *et al.*, 2020] Zhao Kang, Wangtao Zhou, Zhitong Zhao, Junming Shao, Meng Han, and Zenglin Xu. Large-scale multi-view subspace clustering in linear time. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 4412–4419, 2020.

[Liang *et al.*, 2020] Naiyao Liang, Zuyuan Yang, Zhenni Li, Weijun Sun, and Shengli Xie. Multi-view clustering by non-negative matrix factorization with co-orthogonal constraints. *Knowledge-Based Systems*, 194:105582, 2020.

[Liang *et al.*, 2025] Ke Liang, Lingyuan Meng, Hao Li, Jun Wang, Long Lan, Miaomiao Li, Xinwang Liu, and Huaimin Wang. From concrete to abstract: Multi-view clustering on relational knowledge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(10):9043–9060, 2025.

[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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 7576–7584, 2022.

[Liu *et al.*, 2023] Chengliang Liu, Jie Wen, Zhihao Wu, Xiaoling Luo, Chao Huang, and Yong Xu. Information recovery-driven deep incomplete multiview clustering network. *IEEE Transactions on Neural Networks and Learning Systems*, 35(11):15442–15452, 2023.

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

[Liu *et al.*, 2025] Qi Liu, Suyuan Liu, Xinwang Liu, and Jianhua Dai. Latent semantics and anchor graph multilayer learning for multi-view unsupervised feature selection. *IEEE Transactions on Knowledge and Data Engineering*, 37(10):6032–6045, 2025.

[Lu *et al.*, 2024] Jitao Lu, Feiping Nie, Rong Wang, and Xuelong Li. Fast multiview clustering by optimal graph mining. *IEEE Transactions on Neural Networks and Learning Systems*, 35(9):13071–13077, 2024.

[Nie *et al.*, 2014] Feiping Nie, Xiaoqian Wang, and Heng Huang. Clustering and projected clustering with adaptive neighbors. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 977–986, 2014.

[Nie *et al.*, 2016] Feiping Nie, Xiaoqian Wang, Michael Jordan, and Heng Huang. The constrained laplacian rank algorithm for graph-based clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

[Pan and Kang, 2021] ErLin Pan and Zhao Kang. Multi-view contrastive graph clustering. In *Advances in Neural Information Processing Systems*, volume 34, pages 2148–2159, 2021.

[Peng *et al.*, 2022] Zhihao Peng, Hui Liu, Yuheng Jia, and Junhui Hou. Adaptive attribute and structure subspace

clustering network. *IEEE Transactions on Image Processing*, 31:3430–3439, 2022.

[Qin *et al.*, 2025] Yalan Qin, Guorui Feng, and Xinpeng Zhang. Scalable one-pass incomplete multi-view clustering by aligning anchors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 20042–20050, 2025.

[Shao *et al.*, 2016] Weixiang Shao, Lifang He, Chun-ta Lu, and Philip S. Yu. Online multi-view clustering with incomplete views. In *2016 IEEE International Conference on Big Data*, pages 1012–1017, 2016.

[Tan *et al.*, 2023] Yuze Tan, Yixi Liu, Shudong Huang, Wentao Feng, and Jiancheng Lv. Sample-level multi-view graph clustering. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23966–23975, 2023.

[Wang *et al.*, 2022] Siwei Wang, Xinwang Liu, Xinzhong Zhu, Pei Zhang, Yi Zhang, Feng Gao, and En Zhu. Fast parameter-free multi-view subspace clustering with consensus anchor guidance. *IEEE Transactions on Image Processing*, 31:556–568, 2022.

[Wang *et al.*, 2025] Qianqian Wang, Haiming Xu, Zihao Zhang, Wei Feng, and Quanxue Gao. Deep multi-modal graph clustering via graph transformer network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 7835–7843, 2025.

[Wei *et al.*, 2023] Lai Wei, Zhengwei Chen, Jun Yin, Changming Zhu, Rigui Zhou, and Jin Liu. Adaptive graph convolutional subspace clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6262–6271, 2023.

[Wu *et al.*, 2025] Tingting Wu, Zhendong Li, Zhibin Gu, Jiazheng Yuan, and Songhe Feng. KOALA: Kernel coupling and element imputation induced multi-view clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 21581–21589, 2025.

[Xia *et al.*, 2022] Wei Xia, Quanxue Gao, Qianqian Wang, Xinbo Gao, Chris Ding, and Dacheng Tao. Tensorized bipartite graph learning for multi-view clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):5187–5202, 2022.

[Xiao *et al.*, 2023] Shunxin Xiao, Shide Du, Zhaoliang Chen, Yunhe Zhang, and Shiping Wang. Dual fusion-propagation graph neural network for multi-view clustering. *IEEE Transactions on Multimedia*, 25:9203–9215, 2023.

[Xu *et al.*, 2025] HaiMing Xu, Qianqian Wang, Boyue Wang, and Quanxue Gao. Deep fair multi-view clustering with attention kan. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5061–5070, 2025.

[Yang *et al.*, 2024] Zengbiao Yang, Yihua Tan, and Tao Yang. Large-scale multi-view clustering via matrix factorization of consensus graph. *Pattern Recognition*, 155:110716, 2024.

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

[Zhang *et al.*, 2024] Chao Zhang, Deng Xu, Xiuyi Jia, Chunlin Chen, and Huaxiong Li. Continual multi-view clustering with consistent anchor guidance. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, volume 8, pages 5434–5442, 2024.

[Zhang *et al.*, 2025a] Chao Zhang, Zhi Wang, Xiuyi Jia, Zechao Li, Chunlin Chen, and Huaxiong Li. Multi-view clustering with incremental instances and views. *IEEE Transactions on Image Processing*, 34:4203–4214, 2025.

[Zhang *et al.*, 2025b] Guang-Yu Zhang, Dong Huang, and Chang-Dong Wang. Large-scale tensorized multi-view kernel subspace clustering. *ACM Transactions on Intelligent Systems and Technology*, 16(4), 2025.