

# Adaptively-weighted Integral Space for Fast Multiview Clustering

Man-Sheng Chen  
chenmsh27@mail2.sysu.edu.cn  
Sun Yat-sen University  
Guangzhou, China

Tuo Liu  
liut259@mail2.sysu.edu.cn  
Sun Yat-sen University  
Guangzhou, China

Chang-Dong Wang\*  
changdongwang@hotmail.com  
Sun Yat-sen University  
Guangzhou, China

Dong Huang  
huangdonghere@gmail.com  
South China Agricultural University  
Guangzhou, China

Jian-Huang Lai  
stsljh@mail.sysu.edu.cn  
Sun Yat-sen University  
Guangzhou, China

## ABSTRACT

Multiview clustering has been extensively studied to take advantage of multi-source information to improve the clustering performance. In general, most of the existing works typically compute an  $n \times n$  affinity graph by some similarity/distance metrics (e.g. the Euclidean distance) or learned representations, and explore the pairwise correlations across views. But unfortunately, a quadratic or even cubic complexity is often needed, bringing about difficulty in clustering large-scale datasets. Some efforts have been made recently to capture data distribution in multiple views by selecting view-wise anchor representations with  $k$ -means, or by direct matrix factorization on the original observations. Despite the significant success, few of them have considered the view-insufficiency issue, implicitly holding the assumption that each individual view is sufficient to recover the cluster structure. Moreover, the latent integral space as well as the shared cluster structure from multiple insufficient views is not able to be simultaneously discovered. In view of this, we propose an Adaptively-weighted Integral Space for Fast Multiview Clustering (AIMC) with nearly linear complexity. Specifically, view generation models are designed to reconstruct the view observations from the latent integral space with diverse adaptive contributions. Meanwhile, a centroid representation with orthogonality constraint and cluster partition are seamlessly constructed to approximate the latent integral space. An alternate minimizing algorithm is developed to solve the optimization problem, which is proved to have linear time complexity w.r.t. the sample size. Extensive experiments conducted on several real-world datasets confirm the superiority of the proposed AIMC method compared with the state-of-the-art methods.

## CCS CONCEPTS

- Computing methodologies → Machine learning; • Cluster analysis → Multiview clustering.

\*Corresponding author

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MM '22, October 10–14, 2022, Lisboa, Portugal  
© 2022 Association for Computing Machinery.  
ACM ISBN 978-1-4503-9203-7/22/10...\$15.00  
<https://doi.org/10.1145/3503161.3547940>

## KEYWORDS

multiview clustering, large-scale, latent integral space, insufficiency

### ACM Reference Format:

Man-Sheng Chen, Tuo Liu, Chang-Dong Wang, Dong Huang, and Jian-Huang Lai. 2022. Adaptively-weighted Integral Space for Fast Multiview Clustering. In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22), October 10–14, 2022, Lisboa, Portugal*. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3503161.3547940>

## 1 INTRODUCTION

In the big data era, a large quantity of multimedia data from Internet as well as social media spring up, where they are often collected without the label information [7, 25, 33]. Clustering techniques have attracted more and more attention in the unsupervised data mining and machine learning community. Many classical clustering methods have been proposed to automatically explore the correlations between different data objects, such as  $k$ -means as well as its variants, spectral clustering, and subspace clustering. Despite impressive performance, they are merely suitable to single view scenario. In practical applications, data are often represented with diverse features from multiple sources or feature extractors. For instance, a webpage is represented by multiview features based on text, image and video. News can be reported by different languages. Multiple views or feature subsets comprehensively describe different aspects of the data, and it is generally infeasible to just consider single view rather than multiple views. Therefore, multiview clustering is developed to well integrate multiple views or subspace structures to improve the clustering performance.

Considerable efforts have been recently made in the development of multiview clustering to investigate the diverse and complementary information among multiple views [18, 31, 33], in which the existing literatures can be divided into three main categories, i.e., the co-training based models, graph based algorithms and subspace clustering based methods. The co-training based models aim at maximizing the mutual agreement between different views via alternately training [10, 11]. For instance, in [11], Kumar et al. co-regularized the clustering hypotheses such that different representations could agree with each other. The work in [10] proposed to seek for the clusters agreeing across multiple views. Ye et al. studied the underlying clustering by maximizing the sum of weighted affinities between different clusterings of multiple views [30]. The graph based algorithms attempt to combine multiple views for recovering the relationships between views [16, 32]. One of the earliest literatures made bipartite graph construction to link features from

different views [3]. In [16], Nie et al. proposed to simultaneously learn the clustering/semi-supervised classification as well as local structure. The work in [25] coupled the similarity-induced graph associated with multiple views, unified graph and indicator into a unified framework. Supposing that views are generated from the latent subspace, the subspace clustering based methods are developed to learn subspace representations based on the self-expressive property. In [15], Luo et al. simultaneously explored consistency and specificity in multiple subspace representations. The work in [32] proposed a one-step kernel subspace clustering model, in which the shared affinity representation was learned from multiple views. Chen et al. jointly studied the low-rank representation tensor as well as affinity matrix while keeping the local structures [2]. Nevertheless, most of existing works suffer from high computational cost (typically quadratic or even cubic complexity for inevitable  $n \times n$  affinity graph construction), which seriously restricts their efficiency when dealing with large-scale multiview data.

In view of this, some large-scale oriented multiview clustering algorithms have been proposed, which can be mainly divided into two categories, i.e., matrix factorization style algorithms [1, 14, 19, 29] and anchor graph based methods [9, 13]. The matrix factorization style algorithms are able to reduce computational burden without the construction of affinity graph. For instance, Wang et al. attempted to explore the cluster structure with a constrained factor matrix [24]. The work in [6] treated the intermediate factor matrix as a diagonal matrix in 3-factor factorization, improving the efficiency of matrix multiplication in optimization. In [1], Cai et al. efficiently combined heterogeneous factor matrices of large-scale data. Nie et al. [19] proposed to conduct the clustering task simultaneously on row and column of the original data. In [14], Liu et al. unified matrix factorization with partition learning without the non-negativity constraint. However, the efficiency of this kind of algorithms would significantly decrease when facing with large data feature dimension, owing to their direct factorization of the original data observation. The anchor graph based methods often deploy correlations between anchor representation and original data to construct the anchor graph. For instances, the work in [13] achieved the fusion graph construction for multiple anchor graphs via a local manifold fusion strategy. Kang et al. developed an anchor graph-based subspace clustering framework [9], in which view-specific graph was explored with pre-defined anchor representations. The work in [21] attempted to explore anchor graph construction based on the underlying data distribution. Similarly, Wang et al. studied the shared anchor graph with the guidance of consensus anchor representation in multiview data [27]. Despite significant success, the view-insufficiency issue is neglected among them, with the assumption that each individual view is sufficient to recover the cluster partition. Besides, they fail to simultaneously discover the latent integral space from multiple insufficient views as well as the common cluster structure shared by multiview data.

To address the aforementioned problems, in this paper, a novel Adaptively-weighted Integral Space for Fast Multiview Clustering (AIMC) method is developed. Instead of performing direct factorization of the original data observation as the existing matrix factorization based literatures, a latent integral space revealing the cluster structure is focused by considering different adaptive contributions

from multiple views in our framework, which is the enhanced formulation associated with the general fast multiview  $k$ -means and flexibly avoids the curse of dimensionality. To be specific, for each view, we design a view generation model to map the latent integral space into the corresponding view observation while adaptively considering their diverse contributions. Reversely, with input view observations, we can restore the latent integral space by mapping back from given view observations with the corresponding self-conducted confidence. Additionally, a centroid representation with orthogonality constraint and cluster assignment are directly constructed to approximate the latent integral space, which not only focuses on the space itself but also seamlessly associates it with the underlying data partition. For clarity, the flowchart of the proposed AIMC method is illustrated in Figure 1. A four-step alternate minimizing strategy is designed with empirical convergence, in which each variable can be updated efficiently to receive the closed-form solution. Consequently, the reconstruction error and distortion error of data partition are merged in our model to further improve the effectiveness and efficiency of clustering.

The main contributions of our work are summarized as follows:

- Considering the view-insufficiency issue, we study an adaptively weighted integral space for fast multiview clustering, which is coupled with the underlying cluster partition.
- We develop an alternate minimizing strategy to solve the optimization problem, by which the proposed method is proved to have linear time complexity *w.r.t.* the number of samples.
- Extensive experiments are conducted on multiple challenging datasets, including several big datasets, to demonstrate the superiority of the proposed method compared with the state-of-the-art methods.

The rest of this paper is organized as follows. In Section 2, the main notations and basic preliminaries of matrix factorization are provided. The proposed Adaptively-weighted Integral Space for Fast Multiview Clustering method is described in Section 3 in which the optimization algorithm and time computational analysis as well as space complexity analysis are provided. In Section 4, the experimental results are reported, where nine real-world datasets are used and nine state-of-the-art methods are compared. Finally, the paper is concluded in Section 5.

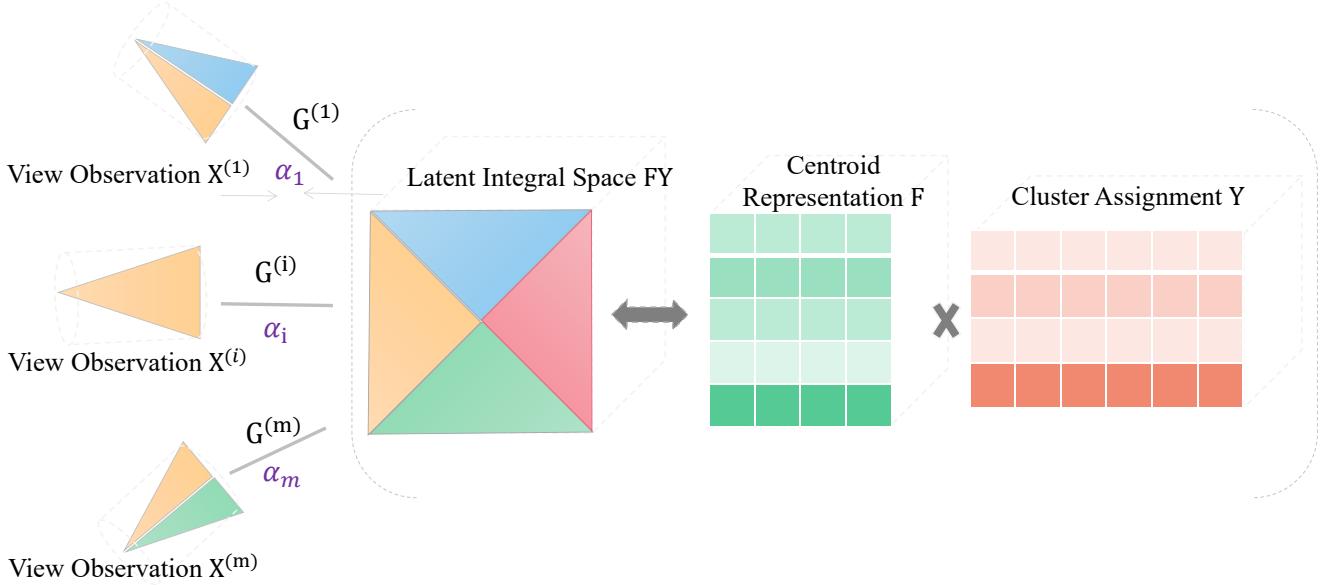
## 2 PRELIMINARIES

### 2.1 Notations

Throughout this paper, matrices are represented as uppercase letters. The  $i$ -th row of matrix  $A$  is denoted as  $A_{i,:}$ , with its  $j$ -th entry being  $A_{ij}$ .  $\|A\|_F^2$  represents the square of Frobenius norm of matrix  $A$ , and  $\|A_{i,:}\|_1$  denotes the  $l_1$  norm of the row vector  $A_{i,:}$ . The trace of  $A$  can be written as  $Tr(A)$ . We denote the  $v$ -th view observation of  $A$  as  $A^{(v)}$ . In particular,  $I_d$  stands for the  $d$  dimensional identity matrix.

### 2.2 Matrix Factorization

Non-negative matrix factorization (NMF) and its variants are often employed in clustering owing to the success in feature extraction. In this part, we will make brief introduction of NMF and one-sided



**Figure 1: Illustration of the proposed AIMC method. The adaptively-weighted latent integral space depicting the common cluster assignment is modeled via view generation models with input view observations carrying diverse adaptive confidences.**

orthogonal NMF [4]. Given the data matrix  $X \in \mathbb{R}^{d \times n}$  drawn from  $k$  partitions, the regularization term about matrix factorization can be generally defined as

$$\min \mathcal{L}(X, WH), \text{ s.t. } W \in C_1, H \in C_2, \quad (1)$$

where  $\mathcal{L}(\cdot)$  stands for the loss function.  $W$  and  $H$  are the factor matrices constrained respectively by  $C_1$  and  $C_2$ .

NMF aims to decompose a non-negative input matrix into two non-negative representation products, i.e., the base matrix and coefficient representation, whose decomposition can be formulated as

$$\min_{W, H} \|X_+ - WH\|_F^2, \text{ s.t. } W \geq 0, H \geq 0, \quad (2)$$

where  $W \in \mathbb{R}^{d \times k}$  is denoted as the base matrix, and  $H \in \mathbb{R}^{k \times n}$  represents the coefficient representation. It can be known that  $H$  is a low-rank coefficient matrix, on which the final cluster partition can be acquired by performing classical clustering algorithm (e.g.  $k$ -means).

The concept of one-sided orthogonal NMF was first proposed by Ding et al, in which the orthogonality constraint was applied to a factor matrix [4]. The objective formulation can be written as

$$\min_{W, H} \|X_+ - WH\|_F^2, \text{ s.t. } WW^T = I_d, H \geq 0, \quad (3)$$

where the base matrix  $W$  becomes an orthogonal matrix. Meanwhile, the formulation above is inherently associated with  $k$ -means algorithm while keeping the orthogonality constraint.

### 3 THE PROPOSED METHOD

In this section, we will first describe the proposed Adaptively-weighted Integral Space for Fast Multiview Clustering (AIMC) model, followed by the detailed optimization procedure. In the meantime, an analysis about the computational time and memory

usage is conducted to demonstrate the time and space efficiency of AIMC.

#### 3.1 Formulation

Since each distinct view representation only consists of partial information about multiple data samples, and the data collection may be noisy, in this paper, we develop a fast multiview clustering framework based on the adaptively-weighted integral space, which depicts the data structure in essence and reveals the common cluster partition shared by different views. As shown in [28], the view specific observation can be generated by mapping from one latent integral space via certain view generation model. Given a multi-view dataset  $\{X^{(v)} \in \mathbb{R}^{d_v \times n}\}_{v=1}^m$  with its  $v$ -th view dimensionality, sample size and view number being respectively  $d_v$ ,  $n$  and  $m$ , the corresponding objective function can be formulated as follow

$$\begin{aligned} & \min_{G^{(v)}, F, Y} \sum_{v=1}^m \|X^{(v)} - G^{(v)}FY\|_F^2, \\ & \text{s.t. } G^{(v)T}G^{(v)} = I_d, F^TF = I_k, Y \in Ind, \end{aligned} \quad (4)$$

where  $G^{(v)} \in \mathbb{R}^{d_v \times d}$  represents the  $v$ -th view generation model.  $F \in \mathbb{R}^{d \times k}$  denotes the centroid representation with its column being the cluster centroid of different partitions, and  $Y \in \mathbb{R}^{k \times n}$  is the cluster assignment matrix with  $Y_{ij} = 1$  if the  $j$ -th instance is assigned to the  $i$ -th cluster and 0 otherwise. Accordingly, the centroid representation and cluster assignment matrix are constructed to approximate the latent integral space, directly ensuring the reliability of view generation model from raw data observation to the underlying data structure. The orthogonality constraint imposed on  $G^{(v)}$  aims to avoid pushing the integral space arbitrarily, while the one about  $F$  attempts to attain mutual independence between different cluster centroid representation.

Despite of this, the model in Eq. (4) treats each view observation equally to discover the integral space, paying no attention to different contributions of multiple views and may resulting in the sub-optimal performance. Therefore, in this paper, an adaptively weighting strategy is designed [18], and the overall objective function can be rewritten as,

$$\begin{aligned} & \min_{G^{(v)}, F, Y} \sum_{v=1}^m \alpha_v \|X^{(v)} - G^{(v)}FY\|_F^2, \\ & \text{s.t. } G^{(v)T}G^{(v)} = I_d, F^TF = I_k, Y \in Ind, \end{aligned} \quad (5)$$

where  $\alpha_v$  is defined as

$$\alpha_v \stackrel{\text{def}}{=} \frac{1}{2\|X^{(v)} - G^{(v)}FY\|_F}. \quad (6)$$

Note that according to Eq. (6),  $\alpha_v$  is dependent on  $G^{(v)}$ ,  $F$  and  $Y$ . Particularly, if view  $v$  is good,  $\|X^{(v)} - G^{(v)}FY\|_F$  should be small, and thus the weight  $\alpha_v$  of view  $v$  would be large according to Eq. (6). Otherwise, small weights would be assigned to poor views, which demonstrates the significance of the self-weighted learning strategy. Further, it is easy to observe that Eq. (5) is equivalent to the following formulation

$$\begin{aligned} & \min_{G^{(v)}, F, Y} \sum_{v=1}^m \|X^{(v)} - G^{(v)}FY\|_F, \\ & \text{s.t. } G^{(v)T}G^{(v)} = I_d, F^TF = I_k, Y \in Ind. \end{aligned} \quad (7)$$

To some extent, the formulation is a multiview least-absolute residual model, which can induce robustness via mitigating the impact of outliers with the present first power residual and achieving dimension reduction by flexible and lower dimension of view generation model during the learning procedure.

### 3.2 Optimization

Due to the non-smoothness and unsupervised learning, it is difficult to handle the problem in Eq. (7). Hence, the following observation is put forward to make it tractable.

**THEOREM 1.** *The model in Eq. (7) is equivalent to the following formulation*

$$\begin{aligned} & \min_{G^{(v)}, F, Y, \alpha_v} \sum_{v=1}^m \alpha_v \|X^{(v)} - G^{(v)}FY\|_F^2, \\ & \text{s.t. } G^{(v)T}G^{(v)} = I_d, F^TF = I_k, Y \in Ind, \alpha \in \Delta_v. \end{aligned} \quad (8)$$

Then, an alternate minimizing algorithm is designed to solve the optimization problem in Eq. (8) by optimizing each variable while fixing the others.

**$G^{(v)}$ -subproblem:** By fixing other variables, the optimization problem for updating  $G^{(v)}$  can be formulated as

$$\min_{G^{(v)}} \sum_{v=1}^m \alpha_v \|X^{(v)} - G^{(v)}FY\|_F^2, \text{ s.t. } G^{(v)T}G^{(v)} = I_d. \quad (9)$$

Since each  $G^{(v)}$  is independent from each other in terms of different views, we can further transform the optimization problem in Eq. (9) into the following model,

$$\max_{G^{(v)}} \text{Tr}(G^{(v)T}H^{(v)}), \text{ s.t. } G^{(v)T}G^{(v)} = I_d, \quad (10)$$

---

**Algorithm 1** Adaptively-weighted Integral Space for Fast Multi-view Clustering

---

**Input:** Multiview dataset  $\{X^{(v)} \in \mathbb{R}^{d_v \times n}\}_{v=1}^m$ , dimension  $d$  and cluster number  $k$ .

- 1: Initialize  $G^{(v)}$ ,  $F$ ,  $Y$ . Initialize  $\alpha_v$  with  $\frac{1}{m}$ .
- 2: **while** not converged **do**
- 3:   Update  $G^{(v)}$ ,  $\forall v$  by solving the problem in Eq. (10).
- 4:   Update  $F$  by solving the problem in Eq. (12).
- 5:   Update  $Y$  via the optimal row obtained by Eq. (15).
- 6:   Update  $\alpha_v$  by solving the problem in Eq. (6).
- 7: **end while**

**Output:** The final consensus cluster assignment matrix  $Y$ .

---

where  $H^{(v)} = X^{(v)}Y^TF^T$ . The optimal solution of  $G^{(v)}$  can be efficiently solved by singular value decomposition (SVD) technique based on  $H^{(v)}$ , and specifically  $G^{(v)}$  can be obtained by calculating  $U_HV_H^T$ , where  $H^{(v)} = U_H\Sigma_HV_H^T$  [26].

**$F$ -subproblem:** By fixing the other variables, the objective function w.r.t.  $F$  can be transformed into the following formulation

$$\min_F \sum_{v=1}^m \alpha_v \|X^{(v)} - G^{(v)}FY\|_F^2, \text{ s.t. } F^TF = I_k. \quad (11)$$

Similar to  $G^{(v)}$ , it is equivalent to solving  $F$  by the following problem

$$\max_F \text{Tr}(F^TJ), \text{ s.t. } F^TF = I_k, \quad (12)$$

where  $J = \sum_{v=1}^m \alpha_v G^{(v)T}X^{(v)}Y^T$ . Therefore, the closed-form solution of  $F$  is equal to  $U_JV_J^T$ , where  $J = U_J\Sigma_JV_J^T$ .

**$Y$ -subproblem:** By fixing other variables,  $Y$  can be updated by solving the following problem

$$\begin{aligned} & \min_Y \sum_{v=1}^m \alpha_v \|X^{(v)} - G^{(v)}FY\|_F^2, \\ & \text{s.t. } Y_{ij} \in \{0, 1\}, \sum_{i=1}^k Y_{ij} = 1, \forall j = 1, 2, \dots, n. \end{aligned} \quad (13)$$

However, it is difficult to directly solve  $Y$  as a whole, since there is one and only one non-zero entry (i.e., 1) in each column of the cluster assignment matrix. Inspired by [5], we can solve the optimization problem in Eq. (13) independently for each object. Therefore, for each object, we can have

$$\begin{aligned} & \min_{Y_{:,j}} \sum_{v=1}^m \alpha_v \|X_{:,j}^{(v)} - G^{(v)}FY_{:,j}\|_F^2, \\ & \text{s.t. } Y_{:,j} \in \{0, 1\}^k, \|Y_{:,j}\|_1 = 1. \end{aligned} \quad (14)$$

We can settle the above optimization problem by finding one optimal row  $i^*$  in  $Y_{:,j}$  such that  $Y_{i^*,j} = 1$  and 0 for other entries in the  $j$ -th column. Then, the optimal row can be searched by

$$i^* = \arg \min_i \sum_{v=1}^m \alpha_v \|X_{:,j}^{(v)} - G^{(v)}F_{:,i}\|_F^2. \quad (15)$$

As a matter of fact, the optimal solution of cluster assignment can be obtained by minimizing the distance between the data sample and the cluster centroid.

**Table 1: Statistics of the nine real-world datasets.**

Datasets	#Object	#View	#Class	View dimension
HandWritten	2000	6	10	216, 76, 64, 6, 240, 47
Notting-Hill	4660	3	5	6750, 3304, 2000
SUNRGBD	10335	2	45	4096, 4096
AwA	30475	6	50	2688, 2000, 252, 2000, 2000, 2000
VGGFace2-50	34027	4	50	944, 576, 512, 640
YTF-10	38654	4	10	944, 576, 512, 640
YTF-20	63896	4	20	944, 576, 512, 640
YTF-50	126054	4	50	944, 576, 512, 640
YTF-100	195537	4	100	944, 576, 512, 640

**$\alpha_v$ -subproblem:** By fixing the other variables except  $\alpha_v$ , the optimization problem is equivalent to dealing with the model in Eq. (6), which is solved by the computed  $G^{(v)}$ ,  $F$  and  $Y$  in each iteration.

According to the convex property and optimal solution of each sub-problem, the objective function in Eq. (8) will decrease monotonically in each iteration until convergence. For clarity, the procedure of solving the proposed AIMC model is summarized in Algorithm 1.

**Remark.** In particular, the view generation model in the proposed AIMC method is exploited to discover the underlying latent integral space, which reflects the data structure as well as cluster partition in essence, and avoids the curse of dimensionality to some degree. To validate the effectiveness of AIMC, we additionally employ a naive orthogonal and non-negative matrix factorization method for multiview clustering (NONMF), whose formulation can be rewritten as

$$\begin{aligned} & \min_{F^{(v)}, Y} \sum_{v=1}^m \|X^{(v)} - F^{(v)}Y\|_F^2, \\ & \text{s.t. } F^{(v)T}F^{(v)} = I_k, Y \in Ind, \end{aligned} \quad (16)$$

where  $F^{(v)}$  is the view-specific cluster centroid matrix with orthogonality constraint, and  $Y$  denotes the shared cluster assignment representation. Due to the direct factorization of original data matrix, the effectiveness and efficiency of NONMF would decrease significantly when the data feature dimension is large. Besides, NONMF can not seek for the underlying data space, failing to take the latent semantics among views into account. More details can be found in Section 4.

### 3.3 Complexity Analysis

**Computational complexity.** The computational burden of AIMC is composed of four parts, i.e., the cost of optimization about each variable. The first part is to update  $G^{(v)}$ , whose cost is  $O(d_v d^2)$  and  $O(d_v dk^2)$  respectively for SVD and matrix multiplication. The second part is about the  $F$  subproblem, which requires  $O(dk^2)$  for performing SVD on  $J$  and  $O(dk^3)$  for matrix multiplication similar to  $G^{(v)}$ . The third part is to compute the final consensus cluster assignment matrix  $Y$ , and  $O(mnk)$  is needed. The last part is to solve the weight  $\alpha_v$ , and it only costs  $O(1)$ . Therefore, the overall computational complexity to solve the proposed AIMC method is  $O((hd^2 + hdk^2 + dk^2 + dk^3 + mnk)t)$ , in which  $h = \sum_{v=1}^m d_v$  is the summation of feature dimensions, and  $t$  stands for the number of iterations of these four parts. Consequently, due to the property of

**Table 2: Complexity analysis on compared methods and AIMC. “Max Reported” represents the largest dataset size reported in the original reference.**

Method	Space Complexity	Time Complexity	Max Reported
AMGL	$O(mn^2 + nk)$	$O(n^3)$	12643
SwMC	$O(n^2 + nk)$	$O(kn^2)$	2000
GMC	$O((m+1)n^2 + nk)$	$O((m+k)n^2)$	11025
PMSC	$O(2mn^2 + (m+1)nk)$	$O(n^3)$	2386
LMVSC	$O(mk(n+h))$	$O(n)$	30,000
SMVSC	$O(mn + (h+m)k)$	$O(n)$	101,499
FPMVS	$O(nk + (h+k)k)$	$O(n)$	101,499
OPMC	$O(nk + (h+k)k)$	$O(n)$	101,499
NONMF	$O(nk + hd)$	$O(n)$	—
AIMC	$O(nk + (h+k)d)$	$O(n)$	195,537

large scale datasets (i.e.,  $n \gg d, k, m$ ), the computational complexity of AIMC is nearly linear to the number of samples  $O(n)$ .

**Space complexity.** The major space usages of the proposed AIMC method are matrices  $G^{(v)} \in \mathbb{R}^{d_v \times d}$ ,  $F \in \mathbb{R}^{d \times k}$ , and  $Y \in \mathbb{R}^{k \times n}$ . Consequently, the major space complexity of AIMC is  $O(nk + (h+k)d)$ , which is nearly linear to the number of samples  $O(n)$  as well.

## 4 EXPERIMENTS

In this section, to verify the effectiveness of the proposed AIMC method, extensive experiments are conducted on nine real-world datasets, including several large-scale multiview image datasets, to compare the proposed AIMC method with four state-of-the-art multiview clustering methods and five large-scale oriented algorithms in terms of clustering performance and running time. In the meantime, parameter analysis and empirical convergence analysis are also conducted for a comprehensive study. All of the experiments are implemented in Matlab 2021a on a standard Window PC with an Intel 2.4-GHz CPU and 64-GB RAM (64-bit).

### 4.1 Datasets and Compared Methods

Nine widely used datasets are adopted in the evaluation, whose statistics are summarized in Table 1. **HandWritten** consists of ten-class handwritten digits, and each digit has 200 samples [22]. In our experiments, six features are selected, which are 216-D profile correlation, 76-D Fourier coefficient, 64-D Karhunen-Loeve coefficient, 6-D morphological feature, 240-D pixel average and 47-D Zernike moment. **Notting-Hill** contains 4660 facial images where the faces of five main casts are selected [34]. In our experiments, three features are selected, including 6750-D Gabor feature, 3304-D LBP feature and 2000-D intensity feature. **SUNRGBD** consists of 10335 indoor scene images of 45 object categories collected from Princeton University [20]. In our experiments, two features are selected, whose dimensions are both 4096. **AwA** is an animal dataset consisting of 30475 images belonging to 50 animal categories [12]. In our experiments, six features are extracted, which are 2688-D color histogram, 2000-D local self-similarity, 252-D PHOG feature, 2000-D SIFT feature, 2000-D color SIFT feature and 2000-D SURF feature. **VGGFace2-50**<sup>1</sup> is a face image dataset containing 34027 images in 50 classes. In our experiments, four features are selected, which are 944-D LBP feature, 576-D HOG feature, 512-D GIST feature and 640-D Gabor feature. **YTF-10** (YouTube Faces), **YTF-20**, **YTF-50** and **YTF-100** are different subsets of face videos obtained from

**Table 3: Comparison results in terms of ACC on nine real-world datasets. N/A implies the out-of-memory error.**

Methods	HandWritten	Notting-Hill	SUNRGBD	AwA	VGGFace2-50	YTF-10	YTF-20	YTF-50	YTF-100
AMGL	0.8082 $\pm$ 0.0566	0.3609 $\pm$ 0.0178	0.2143 $\pm$ 0.0119	N/A	N/A	N/A	N/A	N/A	N/A
SwMC	0.8740 $\pm$ 0.0000	0.3388 $\pm$ 0.0000	0.1181 $\pm$ 0.0000	N/A	N/A	N/A	N/A	N/A	N/A
GMC	0.8820 $\pm$ 0.0891	0.3124 $\pm$ 0.0000	0.1277 $\pm$ 0.0000	N/A	N/A	N/A	N/A	N/A	N/A
PMSC	0.6318 $\pm$ 0.0440	0.7782 $\pm$ 0.0617	N/A						
LMVSC	0.9225 $\pm$ 0.0000	<b>0.9133</b> $\pm$ 0.0000	0.1810 $\pm$ 0.0000	0.0760 $\pm$ 0.0000	0.1113 $\pm$ 0.0000	0.7566 $\pm$ 0.0000	0.7092 $\pm$ 0.0000	0.6825 $\pm$ 0.0000	0.6006 $\pm$ 0.0000
SMVSC	0.8205 $\pm$ 0.0000	0.8081 $\pm$ 0.0000	0.2223 $\pm$ 0.0000	0.0908 $\pm$ 0.0000	0.1127 $\pm$ 0.0000	0.7389 $\pm$ 0.0000	<u>0.7125</u> $\pm$ 0.0000	0.6681 $\pm$ 0.0000	0.5907 $\pm$ 0.0000
FPMLS	0.8230 $\pm$ 0.0000	0.7131 $\pm$ 0.0000	<u>0.2388</u> $\pm$ 0.0000	0.0912 $\pm$ 0.0000	0.1098 $\pm$ 0.0000	0.7325 $\pm$ 0.0000	0.6948 $\pm$ 0.0000	0.6851 $\pm$ 0.0000	0.5293 $\pm$ 0.0000
OPMC	0.8915 $\pm$ 0.0000	0.7431 $\pm$ 0.0000	0.1930 $\pm$ 0.0000	0.0925 $\pm$ 0.0000	<u>0.1160</u> $\pm$ 0.0000	0.7532 $\pm$ 0.0000	0.6782 $\pm$ 0.0000	<u>0.7165</u> $\pm$ 0.0000	0.6443 $\pm$ 0.0000
NONMF	0.8400 $\pm$ 0.0000	0.8027 $\pm$ 0.0000	0.2324 $\pm$ 0.0000	<u>0.0965</u> $\pm$ 0.0000	0.1054 $\pm$ 0.0000	<u>0.7726</u> $\pm$ 0.0000	0.6509 $\pm$ 0.0000	0.7135 $\pm$ 0.0000	<u>0.6610</u> $\pm$ 0.0000
AIMC	<b>0.9345</b> $\pm$ 0.0000	0.9064 $\pm$ 0.0000	<b>0.2536</b> $\pm$ 0.0000	<b>0.1008</b> $\pm$ 0.0000	<b>0.1186</b> $\pm$ 0.0000	<b>0.8881</b> $\pm$ 0.0000	<b>0.7640</b> $\pm$ 0.0000	<b>0.7602</b> $\pm$ 0.0000	<b>0.6789</b> $\pm$ 0.0000

**Table 4: Comparison results in terms of NMI on nine real-world datasets. N/A implies the out-of-memory error.**

Methods	HandWritten	Notting-Hill	SUNRGBD	AwA	VGGFace2-50	YTF-10	YTF-20	YTF-50	YTF-100
AMGL	0.8463 $\pm$ 0.0294	0.0780 $\pm$ 0.0208	0.1686 $\pm$ 0.0109	N/A	N/A	N/A	N/A	N/A	N/A
SwMC	<u>0.8886</u> $\pm$ 0.0000	0.0877 $\pm$ 0.0000	0.0287 $\pm$ 0.0000	N/A	N/A	N/A	N/A	N/A	N/A
GMC	<b>0.9041</b> $\pm$ 0.0000	0.0923 $\pm$ 0.0000	0.0728 $\pm$ 0.0000	N/A	N/A	N/A	N/A	N/A	N/A
PMSC	0.6567 $\pm$ 0.0391	0.6563 $\pm$ 0.0430	N/A						
LMVSC	0.8576 $\pm$ 0.0000	<u>0.7966</u> $\pm$ 0.0000	0.2283 $\pm$ 0.0000	0.0879 $\pm$ 0.0000	0.1332 $\pm$ 0.0000	0.7670 $\pm$ 0.0000	0.7751 $\pm$ 0.0000	0.8098 $\pm$ 0.0000	0.7804 $\pm$ 0.0000
SMVSC	0.7886 $\pm$ 0.0000	0.7371 $\pm$ 0.0000	0.2280 $\pm$ 0.0000	0.1089 $\pm$ 0.0000	<u>0.1366</u> $\pm$ 0.0000	0.7980 $\pm$ 0.0000	0.7913 $\pm$ 0.0000	0.8258 $\pm$ 0.0000	0.7991 $\pm$ 0.0000
FPMLS	0.7923 $\pm$ 0.0000	0.6772 $\pm$ 0.0000	0.2413 $\pm$ 0.0000	0.1088 $\pm$ 0.0000	0.1365 $\pm$ 0.0000	0.7740 $\pm$ 0.0000	0.7790 $\pm$ 0.0000	0.8364 $\pm$ 0.0000	0.7532 $\pm$ 0.0000
OPMC	0.8131 $\pm$ 0.0000	0.7638 $\pm$ 0.0000	<b>0.2523</b> $\pm$ 0.0000	0.1195 $\pm$ 0.0000	0.1344 $\pm$ 0.0000	0.8014 $\pm$ 0.0000	0.7726 $\pm$ 0.0000	0.8278 $\pm$ 0.0000	0.8079 $\pm$ 0.0000
NONMF	0.7981 $\pm$ 0.0000	0.7126 $\pm$ 0.0000	0.2359 $\pm$ 0.0000	0.1173 $\pm$ 0.0000	0.1324 $\pm$ 0.0000	<u>0.8144</u> $\pm$ 0.0000	<u>0.7952</u> $\pm$ 0.0000	<u>0.8479</u> $\pm$ 0.0000	<u>0.8280</u> $\pm$ 0.0000
AIMC	0.8823 $\pm$ 0.0000	<b>0.8226</b> $\pm$ 0.0000	0.2501 $\pm$ 0.0000	<b>0.1200</b> $\pm$ 0.0000	<b>0.1489</b> $\pm$ 0.0000	<b>0.8704</b> $\pm$ 0.0000	<b>0.8334</b> $\pm$ 0.0000	<b>0.8643</b> $\pm$ 0.0000	<b>0.8431</b> $\pm$ 0.0000

**Table 5: Comparison results in terms of Purity on nine real-world datasets. N/A implies the out-of-memory error.**

Methods	HandWritten	Notting-Hill	SUNRGBD	AwA	VGGFace2-50	YTF-10	YTF-20	YTF-50	YTF-100
AMGL	0.8281 $\pm$ 0.0475	0.3677 $\pm$ 0.0161	0.2631 $\pm$ 0.0087	N/A	N/A	N/A	N/A	N/A	N/A
SwMC	0.8795 $\pm$ 0.0000	0.3710 $\pm$ 0.0000	0.1311 $\pm$ 0.0000	N/A	N/A	N/A	N/A	N/A	N/A
GMC	0.8820 $\pm$ 0.0000	0.3380 $\pm$ 0.0000	0.1415 $\pm$ 0.0000	N/A	N/A	N/A	N/A	N/A	N/A
PMSC	0.8311 $\pm$ 0.0363	0.7948 $\pm$ 0.0488	N/A						
LMVSC	0.9225 $\pm$ 0.0000	<b>0.9133</b> $\pm$ 0.0000	0.1856 $\pm$ 0.0000	0.0868 $\pm$ 0.0000	<b>0.1401</b> $\pm$ 0.0000	<u>0.8125</u> $\pm$ 0.0000	0.7614 $\pm$ 0.0000	0.7688 $\pm$ 0.0000	0.6827 $\pm$ 0.0000
SMVSC	0.8205 $\pm$ 0.0000	0.8324 $\pm$ 0.0000	0.3292 $\pm$ 0.0000	0.0966 $\pm$ 0.0000	0.1157 $\pm$ 0.0000	0.7726 $\pm$ 0.0000	0.7700 $\pm$ 0.0000	0.6926 $\pm$ 0.0000	0.6104 $\pm$ 0.0000
FPMLS	0.8230 $\pm$ 0.0000	0.7850 $\pm$ 0.0000	0.3399 $\pm$ 0.0000	0.0963 $\pm$ 0.0000	0.1142 $\pm$ 0.0000	0.7621 $\pm$ 0.0000	0.7259 $\pm$ 0.0000	0.7140 $\pm$ 0.0000	0.5446 $\pm$ 0.0000
OPMC	0.8915 $\pm$ 0.0000	0.8285 $\pm$ 0.0000	<b>0.3974</b> $\pm$ 0.0000	<b>0.1140</b> $\pm$ 0.0000	0.1241 $\pm$ 0.0000	0.8061 $\pm$ 0.0000	0.7250 $\pm$ 0.0000	0.7686 $\pm$ 0.0000	0.6983 $\pm$ 0.0000
NONMF	0.8400 $\pm$ 0.0000	0.8027 $\pm$ 0.0000	0.3327 $\pm$ 0.0000	0.1025 $\pm$ 0.0000	<u>0.1324</u> $\pm$ 0.0000	0.8057 $\pm$ 0.0000	0.7381 $\pm$ 0.0000	0.7531 $\pm$ 0.0000	0.6995 $\pm$ 0.0000
AIMC	<b>0.9345</b> $\pm$ 0.0000	0.9064 $\pm$ 0.0000	0.3519 $\pm$ 0.0000	0.1097 $\pm$ 0.0000	0.1271 $\pm$ 0.0000	<b>0.8881</b> $\pm$ 0.0000	<b>0.8110</b> $\pm$ 0.0000	<b>0.7879</b> $\pm$ 0.0000	<b>0.7267</b> $\pm$ 0.0000

**Table 6: Comparison results in terms of Fscore on nine real-world datasets. N/A implies the out-of-memory error.**

Methods	HandWritten	Notting-Hill	SUNRGBD	AwA	VGGFace2-50	YTF-10	YTF-20	YTF-50	YTF-100
AMGL	0.7681 $\pm$ 0.0853	0.3654 $\pm$ 0.0040	0.1422 $\pm$ 0.0046	N/A	N/A	N/A	N/A	N/A	N/A
SwMC	0.8624 $\pm$ 0.0000	0.3690 $\pm$ 0.0000	0.1204 $\pm$ 0.0000	N/A	N/A	N/A	N/A	N/A	N/A
GMC	<u>0.8653</u> $\pm$ 0.0000	0.3694 $\pm$ 0.0000	0.1215 $\pm$ 0.0000	N/A	N/A	N/A	N/A	N/A	N/A
PMSC	0.5539 $\pm$ 0.0476	0.6568 $\pm$ 0.0565	N/A						
LMVSC	0.8499 $\pm$ 0.0000	<u>0.8347</u> $\pm$ 0.0000	0.1154 $\pm$ 0.0000	0.0377 $\pm$ 0.0000	0.0524 $\pm$ 0.0000	0.7001 $\pm$ 0.0000	0.6268 $\pm$ 0.0000	0.5794 $\pm$ 0.0000	0.5171 $\pm$ 0.0000
SMVSC	0.7525 $\pm$ 0.0000	0.7870 $\pm$ 0.0000	0.1487 $\pm$ 0.0000	<b>0.0641</b> $\pm$ 0.0000	<u>0.0604</u> $\pm$ 0.0000	0.6982 $\pm$ 0.0000	0.6518 $\pm$ 0.0000	0.6157 $\pm$ 0.0000	0.5036 $\pm$ 0.0000
FPMLS	0.7556 $\pm$ 0.0000	0.6928 $\pm$ 0.0000	<u>0.1604</u> $\pm$ 0.0000	0.0640 $\pm$ 0.0000	0.0597 $\pm$ 0.0000	0.6959 $\pm$ 0.0000	0.6261 $\pm$ 0.0000	0.6381 $\pm$ 0.0000	0.3541 $\pm$ 0.0000
OPMC	0.8033 $\pm$ 0.0000	0.7395 $\pm$ 0.0000	0.1339 $\pm$ 0.0000	0.0469 $\pm$ 0.0000	0.0550 $\pm$ 0.0000	0.7225 $\pm$ 0.0000	0.6206 $\pm$ 0.0000	0.6376 $\pm$ 0.0000	0.5651 $\pm$ 0.0000
NONMF	0.7757 $\pm$ 0.0000	0.7802 $\pm$ 0.0000	0.1506 $\pm$ 0.0000	0.0634 $\pm$ 0.0000	0.0580 $\pm$ 0.0000	<u>0.7461</u> $\pm$ 0.0000	0.6590 $\pm$ 0.0000	<u>0.6573</u> $\pm$ 0.0000	0.5463 $\pm$ 0.0000
AIMC	<b>0.8790</b> $\pm$ 0.0000	<b>0.8565</b> $\pm$ 0.0000	<b>0.1709</b> $\pm$ 0.0000	0.0640 $\pm$ 0.0000	<b>0.0606</b> $\pm$ 0.0000	<b>0.8288</b> $\pm$ 0.0000	<b>0.7372</b> $\pm$ 0.0000	<b>0.7004</b> $\pm$ 0.0000	<b>0.6161</b> $\pm$ 0.0000

YouTube<sup>2</sup>, whose data size are respectively 38654, 63896, 126054 and 195537. In our experiments, features of four views are extracted in all these four datasets, including 944-D LBP feature, 576-D HOG feature, 512-D GIST feature and 640-D Gabor feature.

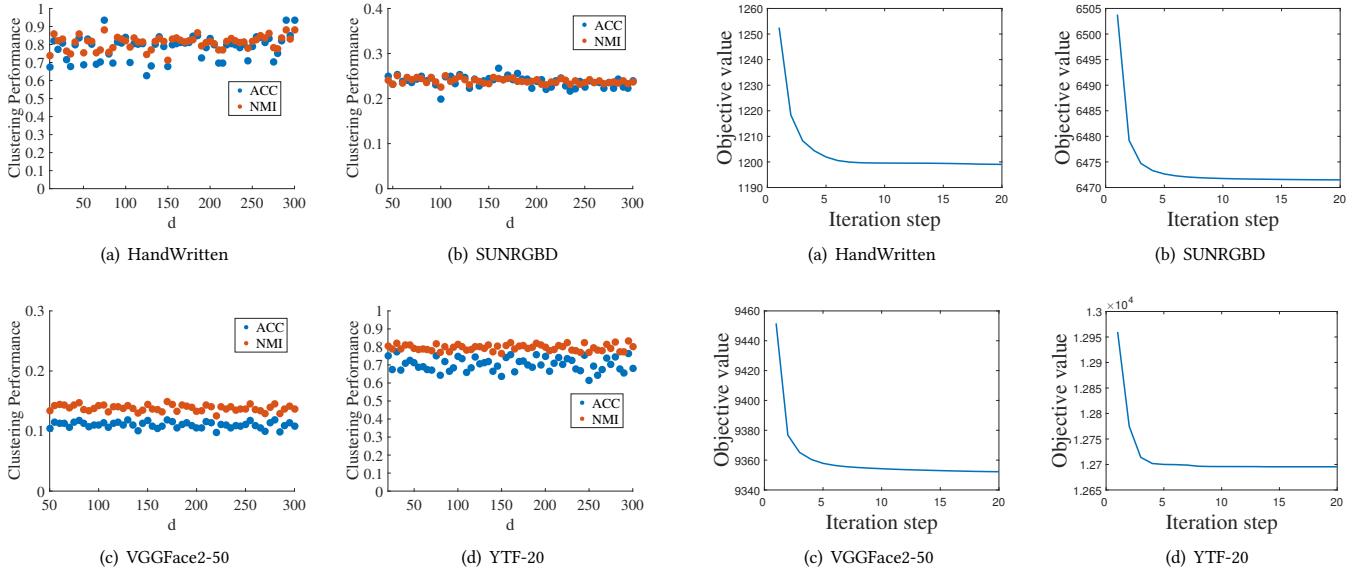
<sup>1</sup>[https://www.robots.ox.ac.uk/~vgg/data/vgg\\_face2/](https://www.robots.ox.ac.uk/~vgg/data/vgg_face2/).

<sup>2</sup><https://www.cs.tau.ac.il/~wolf/ytfaces/>.

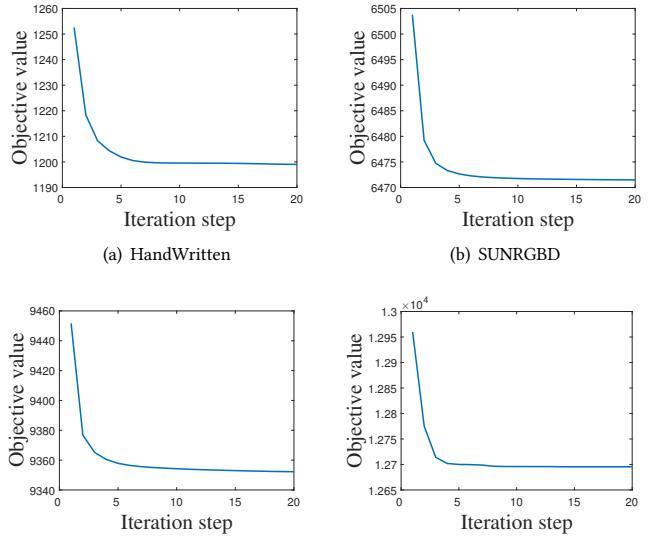
The proposed AIMC method is compared with the following state-of-the-art baselines. **Parameter-free Auto-weighted Multiple Graph Learning (AMGL)** [17] proposes to learn the optimal weights for each graph. **Self-weighted Multiview Clustering with Multiple Graphs (SwMC)** [18] attempts to study a Laplacian rank constrained graph. **Graph-based Multiview Clustering**

**Table 7: Comparison results on nine real-world datasets in terms of running time in seconds. N/A implies the out-of-memory error.**

Datasets	AMGL	SwMC	GMC	PMSC	LMVSC	SMVSC	FPMVS	OPMC	NONMF	AIMC
HandWritten	40.65	144.71	21.75	2133.26	<u>2.20</u>	14.58	21.49	<b>0.97</b>	6.13	2.60
Notting-Hill	592.33	3276.04	27.72	3287.20	<u>2.35</u>	375.08	729.19	<u>20.21</u>	34.45	21.57
SUNRGBD	4504.63	1522.83	483.24	N/A	<b>8.94</b>	551.70	1070.34	145.45	122.22	<u>51.42</u>
AwA	N/A	N/A	N/A	N/A	<b>55.14</b>	2265.28	4025.50	897.73	301.95	<u>166.15</u>
VGGFace2-50	N/A	N/A	N/A	N/A	<b>21.98</b>	820.42	1279.40	256.09	131.42	<u>76.16</u>
YTF-10	N/A	N/A	N/A	N/A	<b>6.78</b>	903.47	1378.40	<u>47.85</u>	60.55	49.91
YTF-20	N/A	N/A	N/A	N/A	<b>15.34</b>	1906.92	2717.5	<u>60.33</u>	153.77	76.84
YTF-50	N/A	N/A	N/A	N/A	<b>119.02</b>	3981.79	7030.41	245.94	284.69	<u>180.49</u>
YTF-100	N/A	N/A	N/A	N/A	<u>345.61</u>	16331.04	19978.28	576.28	404.28	<b>336.32</b>
Rank	8.0	8.7	5.3	10.0	<b>1.2</b>	5.2	6.3	2.9	3.7	<u>2.3</u>

**Figure 2: Parameter analysis: Performance in terms of ACC and NMI with different dimension of the view generation model  $d$  on the four datasets.**

(GMC) [25] learns similarity-induced graphs from multiple views and unified fusion graph in a mutual manner. **Partition Level Multiview Subspace Clustering (PMSC)** [8] develops a unified multiview subspace clustering model. **Large-scale Multiview Subspace Clustering in Linear Time (LMVSC)** [9] proposes a multi-view subspace clustering framework for large-scale data in which a smaller graph for each view is studied. **Scalable Multiview Subspace Clustering with Unified Anchors (SMVSC)** [21] attempts to construct graph representation based on actual latent data distribution. **Fast Parameter-Free Multiview Subspace Clustering With Consensus Anchor Guidance (FPMVS)** [27] studies the subspace representation with consensus anchor guidance. **One-pass Multiview Clustering for Large-scale Data (OPMC)** considers matrix factorization together with partition generation [14]. **Naive Orthogonal and Non-negative Matrix Factorization for multiview clustering (NONMF)** makes direct factorization of

**Figure 3: Convergence analysis: The objective value as a function of the iteration step on the four datasets.**

original data matrix, which is included for ablation study to confirm the superiority of AIMC.

For comparison, we summarize the major space and time complexity of the compared methods in Table 2.

## 4.2 Experimental Setup

For all methods, including the proposed AIMC method and nine compared methods, experiments are conducted twenty times for each of them and the average performance as well as the standard deviation (std. dev.) are reported. In the meantime,  $k$ -means algorithm is needed to achieve the final clustering performance for most of the methods, and thus in each experiment, it is run ten times to eliminate the random initialization. For our method, the dimension of the view generation model  $d$  is tuned from  $[k : 5 : 300]$  for the benchmark datasets, where  $d \geq k$  should be satisfied so as to better describe the cluster space. For the compared methods, the best parameters are tuned as suggested by the corresponding papers.

Comprehensively, the clustering performance is evaluated by four widely used metrics, i.e., accuracy (ACC), normalized mutual information (NMI), purity and Fscore. For all the above evaluation metrics, higher values indicate better clustering performance. More detailed descriptions of the four measures can be referred to [23]. Since each evaluation metric penalizes or favours certain property in clusterings, a more comprehensive evaluation can be received by reporting clustering performance via the four diverse metrics.

### 4.3 Comparison Results

**4.3.1 Clustering Performance.** The clustering performance obtained by different multiview clustering methods on the nine real-world datasets are reported in terms of ACC, NMI, Purity and Fscore in Table 3, 4, 5 and 6, where the mean as well as the standard deviation (std. dev.) are reported over 20 runs. In these tables, the best performance for distinct datasets in terms of each measure is highlighted in boldface, while the second best one in underline. Note that N/A implies that the algorithm suffers out-of-memory error on the corresponding dataset on our device, and thus no result is reported. According to the four tables, the following observations can be obtained.

- (1) In general, the proposed AIMC method achieves the best clustering performance on most of the benchmark datasets, and always obtains better results than other multiview clustering methods for large-scale datasets. For instance, on the VGGFace2-50 dataset, AIMC significantly outperforms the other five large-scale oriented algorithms by achieving improvements of 1.57%, 1.23%, 1.24%, 1.45% and 1.65% in terms of NMI. On the YTF-10 dataset, the performance improvements over the other algorithms in terms of NMI are respectively 10.34%, 7.24%, 9.64%, 6.90% and 5.60%.
- (2) It could be concluded that the anchor based methods (LMVSC, SMVSC, FPMVS) or matrix factorization based methods (OPMC, NONMF, AIMC) are more suitable for large-scale datasets while comparing with the traditional multiview subspace clustering methods (AMGL, SwMC, GMC, PMSC).
- (3) Despite good results achieved by NONMF, the performance of the proposed AIMC method is obviously superior with meaningful discovery of the latent integral space constructed by data cluster structure.

Consequently, empirical comparison results validate the superiority of the proposed AIMC model.

**4.3.2 Running Time.** For further study, the comparison of running time in seconds consumed by all compared algorithms on the nine real-world datasets is reported in Table 7. From the table, it can be observed that a relatively good trade-off between the clustering performance and computational burden is achieved by the proposed AIMC method, which averagely ranks the second best on computational time. Despite less computational time cost by LMVSC, its clustering performance is worse than AIMC on the whole. This is mainly due to that LMVSC merely focuses on the construction of multiple view-specific graph representations, neglecting the cross-view complementary information and underlying correlations from the latent integral space. Note that on the largest benchmark dataset (i.e., YTF-100), the proposed AIMC method needs the least computational cost, indicating the strong potential for facing with bigger

data. As a result, large-scale multiview data clustering problems can be tackled by the proposed AIMC method with efficient computation.

### 4.4 Parameter Analysis

In this subsection, parameter analysis about the dimension of the view generation model  $d$  is conducted on multiple datasets. Due to the page limit, the influences of different parameter values in terms of ACC and NMI on the four randomly selected benchmark datasets, i.e., HandWritten, SUNRGBD, VGGFace2-50 and YTF-20, are illustrated in Figure 2, in which the selection of  $d$  is searched from  $[k : 5 : 300]$  according to different properties of multiple datasets. From the figure, it can be observed that the performance of AIMC is relatively stable over the corresponding ranges of parameter value on different datasets. Meanwhile, we can easily acquire the best clustering performance on different datasets according to the subfigures.

### 4.5 Convergence Analysis

Since the objective formulation in Eq. (8) solved by the alternate minimizing algorithm is non-increasing with iterations, the method can be guaranteed to converge finally. In this subsection, empirical convergence analysis is conducted on the HandWritten, SUNRGBD, VGGFace2-50 and YTF-20 datasets to verify the convergence property of the proposed AIMC method. The objective values on the four benchmark datasets are illustrated in Figure 3. From the figure, we can observe that the corresponding objective value decreases monotonically to a minimum within the first 10 iterations, showing that the proposed model is able to converge within just a few iterations.

## 5 CONCLUSION

In this paper, we develop a novel model termed Adaptively-weighted Integral Space for Fast Multiview Clustering (AIMC) with nearly linear complexity. Within the framework, a view-specific generation model is designed to map the latent integral space into the corresponding view observation with diverse adaptive contributions. Reversely, with input view observations, the latent integral space can be restored by mapping back from given view observations with corresponding self-conducted confidences. In addition, the orthogonal centroid representation and cluster assignment are seamlessly constructed to approximate the latent integral space. By merging the reconstruction error and distortion error of data partition, AIMC is able to simultaneously discover the latent integral space as well as cluster partition. An alternate minimizing algorithm is developed to solve the optimization problem, which is proved to have linear time complexity w.r.t. the sample size. Compared with the state-of-the-art multiview clustering methods and large-scale oriented methods, extensive experimental results on several large-scale datasets have validated the superiority of the proposed method.

## ACKNOWLEDGMENTS

The work was supported by National Key Research and Development Program of China (2021YFF1201200), and NSFC (61876193). Chang-Dong Wang is the corresponding author.

## REFERENCES

- [1] Xiao Cai, Feiping Nie, and Heng Huang. 2013. Multi-View K-Means Clustering on Big Data. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*. 2598–2604.
- [2] Yongyong Chen, Xiaolin Xiao, and Yicong Zhou. 2020. Multi-view subspace clustering via simultaneously learning the representation tensor and affinity matrix. *Pattern Recognit.* 106 (2020), 107441.
- [3] Virginia R De Sa. 2005. Spectral clustering with two views. In *Proceedings of the 22th International Conference on Machine Learning*. 20–27.
- [4] C. Ding, Tao Li, Wei Peng, and Haesun Park. 2006. Orthogonal nonnegative matrix t-factorizations for clustering. In *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 126–135.
- [5] Chris H. Q. Ding and Xiaofeng He. 2005. On the Equivalence of Nonnegative Matrix Factorization and Spectral Clustering. In *Proceedings of the 2005 SIAM International Conference on Data Mining*. 606–610.
- [6] Junwei Han, Kun Song, Feiping Nie, and Xuelong Li. 2017. Bilateral k-Means Algorithm for Fast Co-Clustering. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. 1969–1975.
- [7] Zongmo Huang, Yazhou Ren, Xiaorong Pu, and Lifang He. 2021. Non-Linear Fusion for Self-Paced Multi-View Clustering. In *ACM Multimedia Conference*. 3211–3219.
- [8] Zhao Kang, Xinjia Zhao, Chong Peng, Hongyuan Zhu, Joey Tianyi Zhou, Xi Peng, Wenyu Chen, and Zenglin Xu. 2020. Partition level multiview subspace clustering. *Neural Networks* 122 (2020), 279–288.
- [9] Zhao Kang, Wangtao Zhou, Zhitong Zhao, Junning Shao, Meng Han, and Zenglin Xu. 2020. Large-Scale Multi-View Subspace Clustering in Linear Time. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence*. 4412–4419.
- [10] Abhishek Kumar and Hal Daumé III. 2011. A Co-training Approach for Multi-view Spectral Clustering. In *Proceedings of the 28th International Conference on Machine Learning*. 393–400.
- [11] Abhishek Kumar, Piyush Rai, and Hal Daumé III. 2011. Co-regularized Multi-view Spectral Clustering. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011*. 1413–1421.
- [12] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. 2014. Attribute-Based Classification for Zero-Shot Visual Object Categorization. *IEEE Trans. Pattern Anal. Mach. Intell.* 36, 3 (2014), 453–465.
- [13] Yeqing Li, Feiping Nie, Heng Huang, and Junzhou Huang. 2015. Large-Scale Multi-View Spectral Clustering via Bipartite Graph. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. 2750–2756.
- [14] Jiyuan Liu, Xinwang Liu, Yuexiang Yang, Li Liu, Sisi Wang, Weixuan Liang, and Jiayong Shi. 2021. One-pass Multi-view Clustering for Large-scale Data. In *2021 IEEE/CVF International Conference on Computer Vision*. 12324–12333.
- [15] Shirui Luo, Changqing Zhang, Wei Zhang, and Xiaochun Cao. 2018. Consistent and specific multi-view subspace clustering. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*. 3730–3737.
- [16] Feiping Nie, Guohao Cai, and Xuelong Li. 2017. Multi-View Clustering and Semi-Supervised Classification with Adaptive Neighbours. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. 2408–2414.
- [17] Feiping Nie, Jing Li, and Xuelong Li. 2016. Parameter-Free Auto-Weighted Multiple Graph Learning: A Framework for Multiview Clustering and Semi-Supervised Classification. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. 1881–1887.
- [18] Feiping Nie, Jing Li, and Xuelong Li. 2017. Self-weighted Multiview Clustering with Multiple Graphs. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*. 2564–2570.
- [19] Feiping Nie, Shaojun Shi, and Xuelong Li. 2020. Auto-weighted multi-view co-clustering via fast matrix factorization. *Pattern Recognit.* 102 (2020), 107207.
- [20] Shuran Song, Samuel P. Lichtenberg, and Jianxiong Xiao. 2015. SUN RGB-D: A RGB-D scene understanding benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition*. 567–576.
- [21] Mengjing Sun, Pei Zhang, Siwei Wang, Sihang Zhou, Wenxuan Tu, Xinwang Liu, En Zhu, and Changjian Wang. 2021. Scalable Multi-view Subspace Clustering with Unified Anchors. In *ACM Multimedia Conference*. 3528–3536.
- [22] Martijn van Breukelen, Robert P. W. Duin, David M. J. Tax, and J. E. den Hartog. 1998. Handwritten digit recognition by combined classifiers. *Kybernetika* 34, 4 (1998), 381–386.
- [23] Chang-Dong Wang, Mansheng Chen, Ling Huang, Jian-Huang Lai, and Philip S. Yu. 2021. Smoothness Regularized Multiview Subspace Clustering With Kernel Learning. *IEEE Trans. Neural Networks Learn. Syst.* 32, 11 (2021), 5047–5060.
- [24] Hao Wang, Feiping Nie, Heng Huang, and Fillia Makedon. 2011. Fast Nonnegative Matrix Tri-Factorization for Large-Scale Data Co-Clustering. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*. 1553–1558.
- [25] Hao Wang, Yan Yang, and Bing Liu. 2020. GMC: Graph-Based Multi-View Clustering. *IEEE Trans. Knowl. Data Eng.* 32, 6 (2020), 1116–1129.
- [26] Siwei Wang, Xinwang Liu, En Zhu, Chang Tang, Jiyuan Liu, Jingtao Hu, Jingyuan Xia, and Jianping Yin. 2019. Multi-view Clustering via Late Fusion Alignment Maximization. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*. 3778–3784.
- [27] Siwei Wang, Xinwang Liu, Xinzhang Zhu, Pei Zhang, Yi Zhang, Feng Gao, and En Zhu. 2022. Fast Parameter-Free Multi-View Subspace Clustering With Consensus Anchor Guidance. *IEEE Trans. Image Process.* 31 (2022), 556–568.
- [28] Chang Xu, Dacheng Tao, and Chao Xu. 2015. Multi-View Intact Space Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* 37, 12 (2015), 2531–2544.
- [29] Ben Yang, Xuetao Zhang, Feiping Nie, Fei Wang, Weizhong Yu, and Rong Wang. 2021. Fast Multi-View Clustering via Nonnegative and Orthogonal Factorization. *IEEE Trans. Image Process.* 30 (2021), 2575–2586.
- [30] Yongkai Ye, Xinwang Liu, Jianping Yin, and En Zhu. 2016. Co-regularized kernel k-means for multi-view clustering. In *23rd International Conference on Pattern Recognition*. 1583–1588.
- [31] Chen Zhang, Siwei Wang, Jiyuan Liu, Sihang Zhou, Pei Zhang, Xinwang Liu, En Zhu, and Changwang Zhang. 2021. Multi-view Clustering via Deep Matrix Factorization and Partition Alignment. In *ACM Multimedia Conference*. 4156–4164.
- [32] Guang-Yi Zhang, Yu-Ren Zhou, Xiao-Yu He, Chang-Dong Wang, and Dong Huang. 2020. One-step Kernel Multi-view Subspace Clustering. *Knowledge-Based Systems* 189 (2020).
- [33] Yi Zhang, Xinwang Liu, Siwei Wang, Jiyuan Liu, Sisi Dai, and En Zhu. 2021. One-Stage Incomplete Multi-view Clustering via Late Fusion. In *ACM Multimedia Conference*. 2717–2725.
- [34] Yi-Fan Zhang, Changsheng Xu, Hanqing Lu, and Yeh-Min Huang. 2009. Character identification in feature-length films using global face-name matching. *IEEE Transactions on Multimedia* 11, 7 (2009), 1276–1288.