Anchors Bring Stability and Efficiency: Fast Tensorial Multi-view Clustering on Shuffled Datasets

Jintian Ji
22120385@bjtu.edu.cn
School of Computer Science and Technology,
Beijing Jiaotong University,
Beijing, China

Songhe Feng*
shfeng@bjtu.edu.cn
Tangshan Research Institute,
Beijing Jiaotong University,
Tangshan, Hebei, China

Abstract

Tensorial Multi-view Clustering (TMC) methods have received significant attention due to their ability to capture the high-order correlation among different views. Despite their notable progress, two issues persist: 1) Most TMC methods are highly reliant on prior structure information in sorted datasets, rendering them challenging to handle the shuffled case. 2) Extremely high computational complexity arising from tensor-related operations. To address these limitations, we propose a novel framework termed Anchors Bring Stability and Efficiency: Fast Tensorial Multi-view Clustering on Shuffled Case (SE-FTMC). SE-FTMC first learns a set of anchors and a unified corresponding matrix between anchors and samples, then the low-rank tensor learning is adopted in anchor space instead of sample space to avoid reliance on the prior information hidden in sorted samples and reduce the computational complexity. Finally, SE-FTMC passes the high-order correlations learned in the anchor space into the sample space through the corresponding matrix to improve the efficiency and stability of clustering. Furthermore, SE-FTMC is solved by an efficient algorithm with linear complexity. Extensive experiments on various datasets demonstrate the effectiveness and superiority of our SE-FTMC compared with state-of-the-art methods. The code is publicly available at: https://github.com/jijintian/SE-FTMC.

CCS Concepts

• **Computing methodologies → Cluster analysis**.

Keywords

Tensorial multi-view clustering, shuffled datasets

ACM Reference Format:
Jintian Ji and Songhe Feng. 2018. Anchors Bring Stability and Efficiency: Fast Tensorial Multi-view Clustering on Shuffled Datasets. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX).* ACM, New York, NY, USA, 10 pages. https://doi.org/XXXXXXX.XXXXXXX

*Corresponding author

Conference acronym 'XX, Woodstock, NY

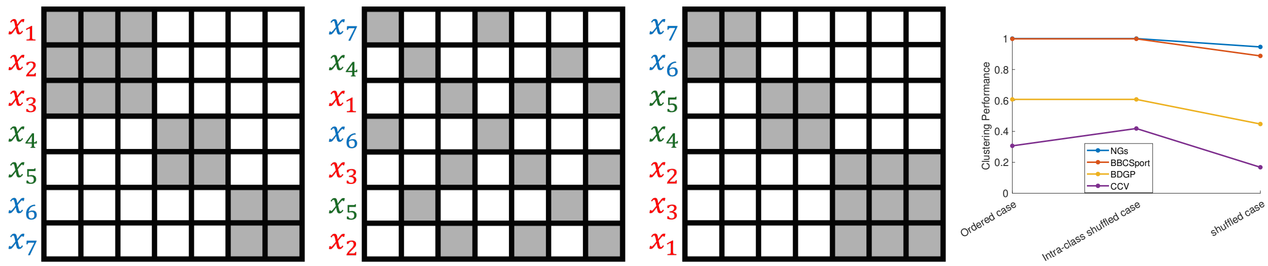
ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

With the rapid development of data collection techniques, objects can be described from multiple sources or views. Multi-view learning aims to leverage the rich information contained in such multi-view data to improve the performance of traditional machine learning tasks. As a fundamental task, multi-view clustering (MVC) [2, 37, 45] aims to partition the samples into different categories by exploiting the consistent and complementary information hidden in multi-view data.

Existing MVC methods can be divided into subspace-based algorithms [3, 12, 41], graph-based algorithms [7, 28, 43], kernel-based algorithms [24, 31, 38], and deep learning based algorithms [6, 8, 49]. In recent years, the low-rank tensor learning framework, which can effectively mine high-order correlations in multi-view data, has been increasingly introduced to improve clustering performance. For example, [50] initially constructs a three-order tensor and introduces a straightforward low-rank constraint to explore the high-order correlations among views. To more effectively constrain the low-rankness of the representation tensor, some works [33, 34, 52] have proposed the Tensor Nuclear Norm (TNN), which is based on the tensor Singular Value Decomposition (t-SVD). However, the t-SVD operation in these methods typically incurs a high computational complexity $O(n^3)$, making the efficiency of the algorithm difficult to guarantee. Many approaches[11, 39, 47] attempt to impose the low-rank constraint on the rotated representation tensor, thereby facilitating a more effective exploration of high-order correlations among views while improving the time complexity to $O(n^2 \log(n))$. Subsequently, some methods employ the anchor learning strategy [14, 16, 46] or tensor factorization strategy [9, 23] to further improve the efficiency of the low-rank tensor learning framework, achieving a time complexity of $O(n \log(n))$.

Although tensor-based multi-view clustering (TMC) methods have achieved remarkable performance in various applications, they suffer from a critical limitation. Most existing algorithms adopt the rotation trick and apply the Fast Fourier Transform (FFT) along the sample mode, making the low-rank learning framework highly dependent on the quality of the input sample structure. These approaches, known as **sample-oriented framework**, are therefore highly sensitive to the structural arrangement of the input data. Specifically, sample-oriented methods perform extremely well when the data exhibit an ordered structure, but their performance degrades significantly when the data structure is altered or shuffled. As illustrated in Figure. 1, input data can typically be arranged in two ways: ordered and shuffled. In the ordered case (Figure. 1 (a)(b)), the samples are grouped according to their categories, preserving a block-diagonal structure in the affinity matrix. This organization



(a) Ordered case (b) Shuffled case (c) Intra-class shuffled case (d) Results of different cases

Figure 1: Input data structure (affinity matrix) of different arrangements. (a).Ordered case: samples are arranged by categories. This type contains prior structural information (i.e., the ideal clustering structure: diagonal block structure). (b). Shuffled case: samples are arranged randomly. This type contains no prior information. (c). Intra-class shuffled case: randomization of samples in the same class without destroying the diagonal block structure. (d). The clustering results of the TMC method (ARLRR [40]*TPAMI*'23) on different arrangements.

provides rich prior structural information, which sample-oriented TMC methods exploit for accurate clustering. In contrast, in the shuffled case (Figure. 1 (b)), the samples are randomly permuted, resulting in a more chaotic affinity matrix with no clear block structure. In real-world scenarios, shuffled data arrangements are far more common than the ideal ordered case. However, most existing TMC algorithms are designed and tested primarily on ordered data, overlooking the challenges posed by shuffled data [27]. This limitation becomes apparent when these methods are applied to shuffled datasets: they suffer from a dramatic loss of accuracy. As demonstrated in Figure. 1 (d), when the input data retains its diagonal block structure (e.g., ordered data or intra-class shuffled data), methods like ARLRR [40] exhibit satisfactory clustering performance. However, when the block structure is lost (the shuffled case), the performance of these TMC methods declines precipitously across all datasets.

This observation reveals a fundamental limitation of the current TMC framework: its inability to maintain robustness under the shuffled case, a scenario frequently encountered in practical clustering tasks. The root cause of this sensitivity lies in the methodological design of existing sample-oriented TMC algorithms, which introduce a rotation trick and apply the FFT along the sample mode. Since the FFT is inherently sensitive to the order of input data, performing this operation on the sample mode makes the algorithm highly dependent on the data arrangement. Consequently, when the input samples are randomly permuted, the performance of these methods degrades significantly, exposing their fragility in unordered or unstructured data environments. In addition to the robustness issue, the current TMC methods suffer from high computational complexity. Owing to the application of low-rank tensor learning in the sample space and the necessity of FFT operations along the sample mode, the overall computational complexity cannot be reduced to linear. Even under ideal conditions, the time complexity remains at least $O(n \log n)$, which poses a substantial bottleneck for scaling these methods to large-scale datasets.

To address the above problems, we propose a novel TMC framework called Anchors Bring Stability and Efficiency: Fast Tensorial Multi-view Clustering on Shuffled Datasets (SE-FTMC). The Figure. 2 illustrates the detailed framework of our proposed model. Specifically, SE-FTMC first learns a set of anchors and the corresponding matrix between anchors and samples, and imposes the low-rank tensor learning framework on the anchor space to mine the high-order correlations among views. Then, the corresponding matrix can pass the information learned in the anchor space to the sample space for clustering purposes. Owing to the optimality that anchors exhibit in the sample space, our proposed SE-FTMC can effectively handle shuffled datasets and exhibit a linear computational complexity. The main contributions of this paper are summarized as follows:

1) SE-FTMC reduces the time complexity of the TMC framework based on rotated tensor to the linear level, thereby significantly enhancing the feasibility of applying low-rank tensor learning frameworks to large-scale multi-view clustering tasks.

2) This paper discusses the problem of the traditional TMC framework that is sensitive to whether the samples are ordered or not, and our proposed SE-FTMC can achieve consistent performance under different cases by adopting the low-rank tensor learning framework in optimal anchor space instead of sample space.

3) An alternating optimization algorithm with good convergence is designed to solve the proposed model. Extensive experiments are conducted to demonstrate the effectiveness of our proposed SE-FTMC compared with state-of-the-art methods.

2 Notations and Related Works

2.1 Notations

The frequently used notations are summarized in Table 1.

Given two three-order tensors $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{B} \in \mathbb{R}^{n_2 \times n_4 \times n_3}$, some tensor-related operations[22] are introduced as fellows:

- Cyclic expansion of the tensor:

$$circ(\mathcal{A}) = \begin{bmatrix} \mathcal{A}^1 & \mathcal{A}^{n_3} & \cdots & \mathcal{A}^2 \\ \mathcal{A}^2 & \mathcal{A}^1 & \cdots & \mathcal{A}^3 \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{A}^{n_3} & \mathcal{A}^{n_3-1} & \cdots & \mathcal{A}^1 \end{bmatrix} \in \mathbb{R}^{n_1 n_3 \times n_2 n_3}. \quad (1)$$

- Tensor unfolding and folding operations:

$$\begin{aligned} &unfold(\mathcal{A}) = \left[\mathcal{A}^1, \mathcal{A}^2, \cdots, \mathcal{A}^{n_3}\right]^T \in \mathbb{R}^{n_1 n_3 \times n_2}. \\ &\mathcal{A} = fold(unfold(\mathcal{A})). \end{aligned} \quad (2)$$

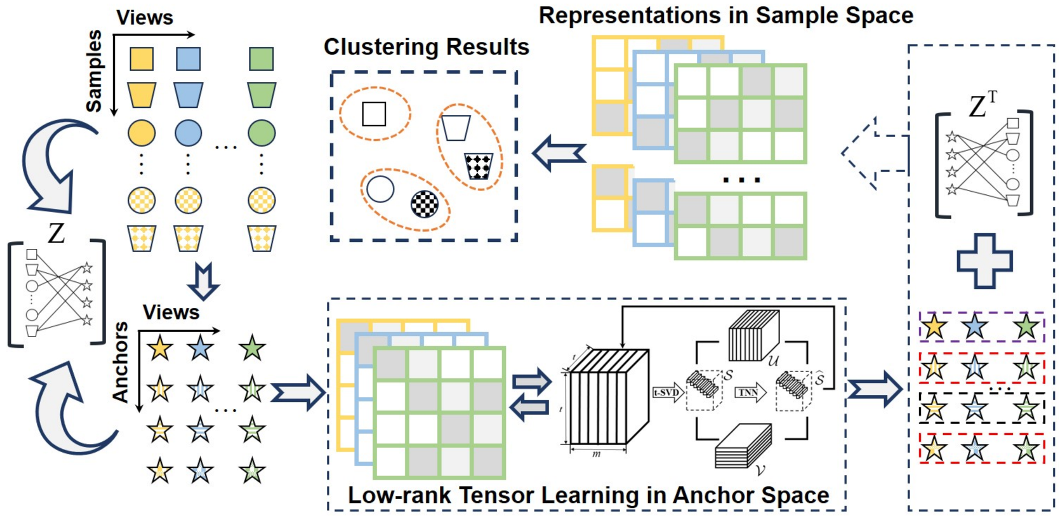


Figure 2: The framework of our proposed SE-FTMC. SE-FTMC employs the low-rank tensor learning framework in the learned anchor space instead of the original sample space to avoid the instability caused by the way the samples are arranged. Then the captured information in the anchor space can pass to the sample space for clustering by the learned corresponding matrix Z.

Table 1: Summary of notations.

Symbol	Definition
x, $\boldsymbol{x}$, $\mathbf{X}$, and $\mathcal{X}$	scalar, vector, matrix, and tensor
$\mathcal{X}^k$	the k-th frontal slice of tensor $\mathcal{X}$
$\mathcal{X}_f = \text{fft}(\mathcal{X}, [], 3)$	the fast Fourier transformation (FFT)
n, t, m	the number of samples, anchors, and views
d^v	the dimension of v-th view
$\mathbf{X}^v \in \mathbb{R}^{n \times d^v}$	the feature matrix of v-th view
$\mathbf{A}^v \in \mathbb{R}^{t \times d^v}$	the anchors of v-th view
$\mathbf{E}^v \in \mathbb{R}^{n \times d^v}$	the reconstruction error in v-th view
$\lVert\cdot\rVert_F$, $\lVert\cdot\rVert_{2,1}$	the Frobenius norm, $\ell_{2,1}$ norm
$\lVert\cdot\rVert_{\circledast}$	the nuclear norm

- t-product operation:

$$\mathcal{A} * \mathcal{B} = fold(circ(\mathcal{A}) \cdot unfold(\mathcal{B})) \in \mathbb{R}^{n_1 \times n_4 \times n_3}. \tag{3}$$

- Tensor Transpose: The transpose of a tensor $\mathcal{A}$ is denoted as $\mathcal{A}^T \in \mathbb{R}^{n_2 \times n_1 \times n_3}$. This operation involves transposing each frontal slice of the tensor $\mathcal{A}$, followed by swapping the second frontal slice with the n_3-th frontal slice.
- f-diagonal Tensor: A tensor $\mathcal{A}$ is called an f-diagonal tensor if and only if each of its frontal slices is a diagonal matrix.

Based on the above tensor operations, the definition of the tensor-Singular Value Decomposition (t-SVD) and Tensor Nuclear Norm (TNN) are defined as follows:

Definition 1. *(t-SVD) [21] Given tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, then the t-SVD of $\mathcal{A}$ is:*

$$\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^T \tag{4}$$

where $\mathcal{U} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$ and $\mathcal{V} \in \mathbb{R}^{n_2 \times n_2 \times n_3}$ are orthogonal tensors, $\mathbf{S} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ *is a f-diagonal tensor.*

Definition 2. *(TNN) [34, 47] Given a tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, then the tensor nuclear norm is defined as:*

$$\lVert\mathcal{A}\rVert_{\circledast} = \frac{1}{n_3} \sum_{k=1}^{n_3} \sum_{i=1}^{h} \mathcal{S}_f^k(i,i) \tag{5}$$

where $h = \min(n_1, n_2)$ and $\mathcal{S}_f$ is from the t-SVD of tensor $\mathcal{A}$ in Fourier domain.

2.2 Tensorial Multi-view Clustering (TMC)

Tensorial Multi-view Clustering (TMC) [4, 17, 19, 36, 47] has recently attracted more attention due to its ability to mine the high-order correlations among views. Depending on the construction of representation matrices, TMC methods can be broadly categorized into graph-based [44, 46, 48], subspace-based [5, 10, 18], and matrix decomposition-based methods [15]. Currently, the subspace-based strategy is a popular approach in TMC. Given a multi-view data set $\{\mathbf{X}^1, \cdots, \mathbf{X}^m\}$ with m views, $\forall v = 1, \cdots, m$, $\mathbf{X}^v \in \mathbb{R}^{n \times d^v}$, the subspace-based TMC methods typically take the following form:

$$\begin{aligned} &\min_{\{\mathbf{Z}_X^v, \mathbf{E}^v\}} \mathcal{T}(\mathcal{Z}_X) + \alpha \mathcal{L}(\{\mathbf{E}^v\}), \\ &s.t.\ \forall v,\ \mathbf{X}^v = \mathbf{Z}_X^v \mathbf{X}^v + \mathbf{E}^v, \mathcal{Z}_X = \Phi(\mathbf{Z}_X^1, \cdots, \mathbf{Z}_X^m), \end{aligned} \tag{6}$$

where α is a trade-off parameter. $\mathcal{T}(\cdot)$ is the low-rank constraint. $\mathbf{E}^v$ means the reconstruction error. $\mathcal{L}(\cdot)$ is designed to measure the error. $\Phi(\cdot)$ denotes merging and rotating operation [47], which merges representation matrices $\mathbf{Z}_X^v \in \mathbb{R}^{n \times n}$ to a three-order tensor $\mathcal{Z}_X$ with the dimension of $n \times m \times n$. This process incurs a complexity of $O(n^2 \log n)$ for learning the low-rank tensor. To reduce the complexity, some methods take the anchor learning strategy [14] ($\mathbf{X}^v = \mathbf{Z}_L^v \mathbf{A}^v + \mathbf{E}^v$, where $\mathbf{A}^v$ means the anchors in v-th view) or matrix decomposition strategy [23] ($\mathbf{X}^v = \mathbf{Z}_L^v \mathbf{H}^v + \mathbf{E}^v$, where $\mathbf{H}^v$ means the base matrix in v-th view.) to reduce the representation matrices $\mathbf{Z}_X^v \in \mathbb{R}^{n \times n}$ into low-dimensional representation matrices

$\mathbf{Z}_L^v \in \mathbb{R}^{n\times h}$, where h means the number of anchors or a small dimension. Consequently, these methods achieve a time complexity of $O(n\log(n))$ for processing the tensor constructed from $\mathbf{Z}_L^v$.

However, current sample-oriented TMC methods still have two main limitations. First, due to the adoption of the rotation trick, the FFT operation is applied along the sample mode, making the method highly sensitive to the ordering of input samples. As a result, these algorithms exhibit satisfactory performance only when the data is presented in a specific, ordered arrangement, but their effectiveness degrades significantly when faced with shuffled or randomly ordered datasets. Second, the imposition of the low-rank tensor learning in the sample space limits the algorithm's complexity to a lower bound of $O(n\log(n))$, preventing TMC methods from achieving linear complexity and thereby restricting their scalability for large datasets.

3 Proposed Method

As mentioned above, the limitations of current TMC methods are caused by their sample-oriented framework. This strategy makes them sensitive to sample arrangement and limits their scalability to large-scale datasets. Being aware of these, we propose to employ the low-rank tensor learning framework in the more stable anchor space, based on Assumption 1. It is noted that the arrangement of samples in the multi-view data does not affect the anchors, so the low-rank tensor learning framework imposed on the anchor space can achieve more stable performance than that in the sample space, especially when the samples are randomly shuffled.

Assumption 1. (Anchor Assumption) *Given a dataset* $\mathbf{X} = [\boldsymbol{x}_1;\cdots;\boldsymbol{x}_n]$ *comprising n samples, the concept of anchors* $\mathbf{A} = [\boldsymbol{a}_1;\cdots;\boldsymbol{a}_t]$ *refers to a set of representative samples or points that could represent a subgroup of samples or cover their data cloud. The anchors are expected to possess the following characteristics:*

1) Samples belonging to the same category may require more than one anchor to cover.

2) The arrangement of the samples does not influence the anchors in the sample space.

Rather than adopting the pre-defined anchor strategy [20, 25, 46], we employ the anchor learning strategy to learn a set of optimal anchors to avoid the sub-optimal problem. Given a multi-view data set $\{\mathbf{X}^1,\cdots,\mathbf{X}^m\}, \forall v, \mathbf{X}^v \in \mathbb{R}^{n\times d^v}$ with m views, where d^v is the dimension of the v-th view, the anchor learning strategy can be expressed as the following problem,

$$\begin{aligned}&\min_{\{\mathbf{A}^v,\mathbf{Z},\mathbf{E}^v\}} \mathcal{L}(\{\mathbf{E}^v\})\\ &s.t.\ \forall v,\ \mathbf{X}^v = \mathbf{Z}\mathbf{A}^v + \mathbf{E}^v, \mathbf{A}^v\mathbf{A}^{vT} = \mathbf{I},\end{aligned} \tag{7}$$

where $\mathcal{L}(\cdot)$ is designed to to measure the reconstruction error, $\mathbf{A}^v \in \mathbb{R}^{t\times d^v}$ means the anchor set in v-th view, with t representing the number of anchors. According to Assumption 1, $c \le t$, with c denoting the number of clusters. $\mathbf{Z} \in \mathbb{R}^{n\times t}$ serves as the unified corresponding matrix, facilitating information exchange between the sample space and the anchor space. Compared to other methods [30, 42, 51] that also use the anchor learning strategy, these algorithms typically leverage the learned anchors to project the samples into a lower-dimensional space, facilitating efficient cross-view information capture in the sample space. In contrast, our approach treats the learned anchors as a new set of samples, enabling the learning of multi-view consistency and complementary information in a more compact and efficient anchor space. So given the learned anchors $\{\mathbf{A}^v\}_{v=1}^m$, the low-rank tensor learning framework can be imposed in anchor space to mine the high-order correlations among views,

$$\begin{aligned}&\min_{\{\mathbf{G}^v,\mathbf{E}^v\}} \|\mathcal{G}\|_{\circledast} + \alpha\|\mathbf{E}\|_{2,1}\\ &s.t.\ \forall v,\ \mathbf{A}^v = \mathbf{G}^v\mathbf{A}^v + \mathbf{E}^v,\\ &\quad \mathbf{E} = [\mathbf{E}^1,\cdots,\mathbf{E}^m]^T, \mathcal{G} = \Phi(\mathbf{G}^1,\cdots,\mathbf{G}^m),\end{aligned} \tag{8}$$

where $\mathbf{G}^v \in \mathbb{R}^{t\times t}$ denotes the representation matrices of anchors, $\Phi(\cdot)$ merges representation matrices $\mathbf{G}^v$ to a three-order tensor $\mathcal{G}$ with the dimension of $t\times m\times t$, such tensor enjoys a $t\log(t)$ complexity in tensor-related operations, which is much lower than $n\log(n)$ in previous TMC works[16, 46].

Combining Eq. (7) and Eq. (8), we formulate the objective function of SE-FTMC as follows:

$$\begin{aligned}&\min_{\{\mathbf{Z},\mathcal{G},\mathbf{A}^v,\mathbf{E}\}} \|\mathcal{G}\|_{\circledast} + \alpha\|\mathbf{E}\|_{2,1}\\ &s.t.\ \forall v,\ \mathbf{X}^v = \mathbf{Z}\mathbf{G}^v\mathbf{A}^v + \mathbf{E}^v,\ \mathbf{Z}^T\mathbf{Z} = \mathbf{I},\\ &\quad \mathbf{A}^v\mathbf{A}^{vT} = \mathbf{I},\ \mathbf{E} = [\mathbf{E}^1,\cdots,\mathbf{E}^m]^T,\\ &\quad \mathcal{G} = \Phi(\mathbf{G}^1,\cdots,\mathbf{G}^m).\end{aligned} \tag{9}$$

When all variables reach their optimal values, we can directly perform k-means on $\mathbf{Z}$ to obtain the clustering results.

4 Optimization

Inspired by the alternating direction method of multipliers (ADMM) [26], the optimal problem Eq. (9) can be solved through an alternating optimization approach. We first introduce an auxiliary tensor variable $\mathcal{J}$ and rewrite Eq. (9) as the following unconstrained problem. Then, we update each one individually while keeping the others fixed.

$$\begin{aligned}&\mathcal{L}(\mathbf{Z},\{\mathbf{G}^v\}_{v=1}^m,\{\mathbf{A}^v\}_{v=1}^m,\mathcal{J},\mathbf{E},\{\mathbf{Y}^v\}_{v=1}^m,\mathcal{W})\\ &= \|\mathcal{J}\|_{\circledast} + \alpha\|\mathbf{E}\|_{2,1} + \langle\mathcal{W},\mathcal{G}-\mathcal{J}\rangle\\ &+ \sum_{v=1}^m \big(\langle\mathbf{Y}^v,\mathbf{X}^v - \mathbf{Z}\mathbf{G}^v\mathbf{A}^v - \mathbf{E}^v\rangle\\ &+ \frac{\mu}{2}\left\|\mathbf{X}^v - \mathbf{Z}\mathbf{G}^v\mathbf{A}^v - \mathbf{E}^v\right\|_F^2\big) + \frac{\rho}{2}\|\mathcal{G}-\mathcal{J}\|_F^2,\end{aligned} \tag{10}$$

4.1 Z-Problem:

Fixing other variables, we can update $\mathbf{Z}$ by solving following problem:

$$\mathbf{Z}^* = \underset{\mathbf{Z}^T\mathbf{Z}=\mathbf{I}}{\arg\max}\, Tr(\mathbf{Z}^T\mathbf{N}), \tag{11}$$

where $\mathbf{N} = \sum_{v=1}^m(\mathbf{Y}^v + \mu\mathbf{X}^v - \mu\mathbf{E}^v)\mathbf{A}^{vT}\mathbf{G}^{vT}$. The optimal solution of $\mathbf{Z}$ is $\mathbf{U}_Z\mathbf{V}_Z^T$, where $\mathbf{U}_Z$ and $\mathbf{V}_Z$ are the left and right singular matrix of $\mathbf{N}$.

4.2 $\mathbf{G}^v$-Problem:

The subproblem of $\mathbf{G}^v$ can be rewritten as:

$$\begin{aligned}\min_{\mathbf{G}^v} &\langle \mathcal{W}, \mathcal{G}-\mathcal{J}\rangle + \langle \mathbf{Y}^v, \mathbf{X}^v - \mathbf{Z}\mathbf{G}^v\mathbf{A}^v - \mathbf{E}^v\rangle \\ &+ \frac{\mu}{2}\left\|\mathbf{X}^v - \mathbf{Z}\mathbf{G}^v\mathbf{A}^v - \mathbf{E}^v\right\|_F^2 + \frac{\rho}{2}\left\|\mathcal{G}-\mathcal{J}\right\|_F^2\end{aligned} \tag{12}$$

This problem admits a closed-form solution by directly applying the first-order optimality condition:

$$\mathbf{G}^v = \frac{1}{\mu+\rho}\mathbf{Z}^T(\mathbf{Y}^v + \mu\mathbf{X}^v - \mu\mathbf{E}^v)\mathbf{A}^{vT} + \rho\mathbf{J}^v - \mathbf{W}^v \tag{13}$$

4.3 E-Problem:

The sub-problem of $\mathbf{E}$ can be rephrased as,

$$\arg\min_{\mathbf{E}} \frac{\alpha}{\mu}\|\mathbf{E}\|_{2,1} + \frac{1}{2}\left\|\mathbf{E}-\hat{\mathbf{E}}\right\|_F^2, \tag{14}$$

where $\hat{\mathbf{E}}$ is constructed by horizontally concatenating the matrices $(\mathbf{X}^v - \mathbf{Z}\mathbf{G}^v\mathbf{A}^v - \mathbf{E}^v + \frac{1}{\mu}\mathbf{Y}^v)^T$ together along row. From the $\ell_{2,1}$ minimization thresholding operator in [32], we can obtain solution as follows:

$$\mathbf{E}_{:,j} = \begin{cases} \dfrac{\left\|\hat{\mathbf{E}}_{:,j}\right\|_2 - \frac{\alpha}{\mu}}{\left\|\hat{\mathbf{E}}_{:,j}\right\|_2}\hat{\mathbf{E}}_{:,j}, & \left\|\hat{\mathbf{E}}_{:,j}\right\|_2 > \dfrac{\alpha}{\mu}, \\ 0, & \text{otherwise.}\end{cases} \tag{15}$$

where $\hat{\mathbf{E}}_{:,j}$ is the j-th column of $\hat{\mathbf{E}}$.

4.4 $\mathcal{J}$-Problem:

When other variables are fixed, the subproblem for $\mathcal{J}$ is formulated as,

$$\arg\min_{\mathcal{J}} \frac{1}{\rho}\|\mathcal{J}\|_{\circledast} + \frac{1}{2}\left\|\mathcal{J} - (\mathcal{G} + \frac{\mathcal{W}}{\rho})\right\|_F^2. \tag{16}$$

Problem Eq. (16) is a typical t-SVD-based tensor nuclear norm minimization problem. The closed-form solution of this optimization problem can be obtained through [47].

4.5 $\mathbf{A}^v$-Problem:

Fixing the other variables, $\mathbf{A}^v$ can be updated by,

$$\mathbf{A}^{v*} = \arg\max_{\mathbf{A}^v\mathbf{A}^{vT}=\mathbf{I}} Tr(\mathbf{A}^v\mathbf{M}), \tag{17}$$

where $\mathbf{M} = (\mathbf{Y}^v + \mu\mathbf{X}^v - \mu\mathbf{E}^v)^T\mathbf{Z}\mathbf{G}^v$. The optimal solution of $\mathbf{A}^v$ is $\mathbf{V}_A\mathbf{U}_A{}^T$, where $\mathbf{U}_A$ and $\mathbf{V}_A$ are the left and right singular matrix of $\mathbf{M}$.

At last, the Lagrange multipliers and penalty parameters are updated as follows,

$$\begin{cases}\mathbf{Y}^v = \mathbf{Y}^v + \mu(\mathbf{X}^v - \mathbf{Z}\mathbf{G}^v\mathbf{A}^v - \mathbf{E}^v) \\ \mathcal{W} = \mathcal{W} + \rho(\mathcal{G}-\mathcal{J}) \\ \mu = \eta_\mu\mu, \rho = \eta_\rho\rho\end{cases} \tag{18}$$

where $\eta_\mu, \eta_\rho > 1$ are used to accelerate convergence. The complete procedure is summarized in Algorithm 1.

Algorithm 1 Optimization Algorithm of Eq. (9)

Input: Multi-view data $\{\mathbf{X}^1, \dots, \mathbf{X}^m\}$, cluster number c, trade-off parameter α.
Initialize: Set $\mathbf{A}^v, \mathbf{Z}, \mathbf{E}^v, \mathbf{Y}^v$ to zero matrix , $\mathcal{J} = \mathcal{W} = \mathbf{0}$;
1: **while** not converge **do**
2: Update $\mathbf{Z}$ by Eq. (11);
3: Update $\mathbf{G}^v$ by Eq. (13);
4: Update $\mathbf{E}$ by Eq. (15);
5: Update $\mathcal{J}$ by Eq. (16);
6: Update $\mathbf{A}^v$ by Eq. (17);
7: Update μ, ρ, $\mathbf{Y}^v$, and $\mathcal{W}$ by Eq.(18);
8: Check the convergence conditions: $\|\mathbf{X}^v - \mathbf{Z}\mathbf{G}^v\mathbf{A}^v - \mathbf{E}^v\|_\infty < \epsilon$ & $\|\mathcal{G}-\mathcal{J}\|_\infty < \epsilon$
9: **end while**
10: Perform k-means on $\mathbf{Z}$ to obtain the final clustering results.

4.6 Computational Complexity Analysis

In this section, we analyze both the time complexity and the space complexity of our proposed SE-FTMC.

Time Complexity: In our proposed SE-FTMC, the time consumption mainly focuses on the update of each variable. Updating $\mathbf{Z}$ requires $O(ntd + t^2d)$, where $d = \sum_{v=1}^m d^v$. For $\mathbf{G}^v$, it needs $O(ntd^v + t^2d^v + t^3)$ operations. Updating $\mathbf{E}$ only costs $O(nd)$. For tensor $\mathcal{J}$, the updating rule involves FFT, inverse FFT, and t-SVD operators, accumulating a complexity of $O(mt^2\log t + m^2t^2)$. For $\mathbf{A}^v$, the time complexity is $O(nt^2 + ntd^v)$. Considering the fact that $t, m \ll n$, the overall time complexity of our proposed SE-FTMC is $O(n)$, which is linear to the number of samples. Compared to the time complexity $O(n^2\log n)$ in the traditional tensorial multi-view clustering methods, SE-FTMC can address large-scale datasets efficiently.

Space Complexity: The main space cost is the matrices of variables: $\mathbf{Z} \in \mathbb{R}^{n\times t}$, $\mathbf{G}^v \in \mathbb{R}^{t\times t}$, $\mathbf{E} \in \mathbb{R}^{n\times d}$, $\mathcal{J} \in \mathbb{R}^{t\times m\times t}$, $\mathbf{A}^v \in \mathbb{R}^{t\times d^v}$ and the matrix multiplication results. Since $t, m \ll n$, the space complexity of SE-FTMC is also linear to the number of samples, $O(n)$.

Table 2: Details of the used datasets.

Dataset	Type	Sample &Cluster	View
NGs	Text	500 & 5	3
BBCSport	Text	544 &5	2
HW	Figure	2000 & 10	6
BDGP	Genome	2500 & 5	2
CIFAR10	Object	50000 & 10	3
CIFAR100	Object	50000 & 100	3
YoutubeFace	Video	101499 & 31	5

5 Experiment

In this section, to verify the effectiveness of SE-FTMC, we compare SE-FTMC with some state-of-the-art clustering methods on several widely used benchmark datasets. The experiments are implemented on a computer with a 2.50GHz i7-11700 CPU and 32GB RAM, MATLAB R2021a.

Table 3: Results of our proposed method and other compared methods on four shuffled datasets.

Methods	Complexity		ACC(%)	NMI(%)	PUR(%)	F-score(%)	ARI(%)	Running time(s)
	Time	Space						
NGs-Shuffled								
LVMSC (AAAI 2020)	$O(n)$	$O(n)$	39.92(1.35)	18.65(5.05)	60.40(16.12)	33.22(3.63)	10.50(1.67)	1.26(0.12)
AWMVC (AAAI 2023)	$O(n)$	$O(n^2)$	44.84(0.68)	31.35(0.52)	48.03(0.55)	38.22(0.40)	17.05(0.19)	3.88(0.61)
CAMVC (AAAI 2024)	$O(n)$	$O(n)$	98.81(0.14)	96.04(0.44)	98.81(0.14)	97.63(0.28)	97.04(0.35)	0.33(0.17)
3AMVC (ACM MM 2024)	$O(n)$	$O(n)$	45.44(0.79)	29.05(1.08)	49.72(1.27)	39.90(0.47)	22.10(1.19)	0.51(0.03)
ASR-ETR (ICCV 2023)	$O(n\log(n))$	$O(n)$	95.36(0.33)	86.17(0.78)	95.36(0.33)	90.93(0.64)	88.86(0.79)	3.60(0.41)
TLSpNM (TPAMI 2023)	$O(n^2\log(n))$	$O(n^2)$	98.80(0.00)	95.82(0.00)	98.80(0.00)	97.60(0.00)	97.01(0.00)	25.08(1.78)
ARLRR (TPAMI 2023)	$O(n^2\log(n))$	$O(n^2)$	97.27(0.12)	91.31(0.30)	97.27(0.12)	94.60(0.23)	93.26(0.29)	6.69(0.46)
OURS	$O(n)$	$O(n)$	98.84(0.08)	96.16(0.30)	98.84(0.08)	97.68(0.18)	97.11(0.23)	2.02(0.45)
BBCSport-Shuffled								
LVMSC (AAAI 2020)	$O(n)$	$O(n)$	64.04(7.28)	47.72(6.55)	74.26(6.11)	53.33(9.20)	35.97(13.14)	0.56(0.41)
AWMVC (AAAI 2023)	$O(n)$	$O(n^2)$	58.40(0.65)	43.13(1.27)	66.35(1.53)	46.74(1.11)	30.81(1.83)	3.53(0.45)
CAMVC (AAAI 2024)	$O(n)$	$O(n)$	91.55(1.59)	81.90(1.61)	92.05(1.24)	84.85(2.06)	80.15(2.64)	0.64(0.36)
3AMVC (ACM MM 2024)	$O(n)$	$O(n)$	53.55(10.61)	30.24(10.95)	57.60(9.07)	40.86(6.97)	21.39(9.59)	4.76(0.11)
ASR-ETR (ICCV 2023)	$O(n\log(n))$	$O(n)$	83.27(0.37)	67.02(1.34)	83.27(0.37)	74.14(2.69)	66.17(3.24)	3.68(0.32)
TLSpNM (TPAMI 2023)	$O(n^2\log(n))$	$O(n^2)$	75.77(5.79)	54.03(4.38)	77.90(2.54)	68.70(3.58)	55.51(6.13)	25.88(1.25)
ARLRR (TPAMI 2023)	$O(n^2\log(n))$	$O(n^2)$	88.85(0.11)	77.84(0.16)	88.85(0.11)	85.25(0.23)	80.59(0.29)	5.69(0.38)
OURS	$O(n)$	$O(n)$	92.94(1.08)	81.30(2.58)	92.94(1.08)	85.78(1.80)	81.28(2.53)	2.35(0.14)
BDGP-Shuffled								
LVMSC (AAAI 2020)	$O(n)$	$O(n)$	53.72(1.53)	30.64(2.80)	55.74(3.51)	40.92(2.37)	25.23(3.50)	2.35(0.14)
AWMVC (AAAI 2023)	$O(n)$	$O(n^2)$	47.88(0.44)	28.78(0.67)	50.51(0.29)	39.88(0.35)	24.66(0.49)	4.31(1.53)
CAMVC (AAAI 2024)	$O(n)$	$O(n)$	51.93(1.65)	28.57(0.88)	52.36(1.58)	42.85(0.98)	26.52(1.27)	1.23(0.33)
3AMVC (ACM MM 2024)	$O(n)$	$O(n)$	46.39(3.18)	28.96(2.05)	49.85(1.26)	36.67(0.99)	17.25(1.23)	5.79(1.24)
ASR-ETR (ICCV 2023)	$O(n\log(n))$	$O(n)$	52.18(4.66)	28.86(2.54)	52.40(4.96)	38.89(3.15)	23.10(4.09)	3.31(0.19)
TLSpNM (TPAMI 2023)	$O(n^2\log(n))$	$O(n^2)$	54.34(0.80)	36.21(0.56)	59.93(0.81)	43.68(0.78)	30.81(0.97)	144.49(3.56)
ARLRR (TPAMI 2023)	$O(n^2\log(n))$	$O(n^2)$	45.56(2.15)	24.19(2.58)	48.39(2.56)	37.45(1.54)	18.06(3.50)	222.10(6.78)
OURS	$O(n)$	$O(n)$	60.90(1.41)	39.31(1.55)	61.06(1.16)	45.74(0.74)	31.53(1.15)	4.01(0.28)
HW-Shuffled								
LVMSC (AAAI 2020)	$O(n)$	$O(n)$	88.86(2.32)	82.28(2.24)	88.86(2.32)	88.91(4.31)	76.50(4.84)	10.42(1.03)
AWMVC (AAAI 2023)	$O(n)$	$O(n^2)$	81.18(3.98)	77.28(2.73)	82.67(3.72)	73.87(4.03)	70.90(4.50)	0.52(0.02)
CAMVC (AAAI 2024)	$O(n)$	$O(n)$	90.09(1.07)	84.30(0.78)	90.28(0.96)	82.98(1.27)	81.08(0.39)	1.21(0.29)
3AMVC (ACM MM 2024)	$O(n)$	$O(n)$	89.82(1.91)	86.33(2.22)	90.52(1.62)	84.02(2.66)	82.20(2.96)	10.16(1.54)
ASR-ETR (ICCV 2023)	$O(n\log(n))$	$O(n)$	82.94(2.85)	73.89(1.66)	82.94(2.85)	71.72(2.65)	68.56(2.93)	2.54(0.69)
TLSpNM (TPAMI 2023)	$O(n^2\log(n))$	$O(n^2)$	92.10(0.80)	85.70(1.10)	92.10(0.80)	84.97(1.38)	83.67(1.49)	238.27(4.56)
ARLRR (TPAMI 2023)	$O(n^2\log(n))$	$O(n^2)$	87.68(0.10)	80.44(0.21)	87.68(0.10)	78.28(0.17)	75.87(0.19)	382.24(57.32)
OURS	$O(n)$	$O(n)$	93.46(0.68)	87.94(0.84)	93.46(0.68)	87.52(1.17)	86.14(1.30)	1.23(0.11)

5.1 Experimental Setup

Datasets: We select seven challenging datasets to conduct the experiments, including NGs[1], BBCSport[2], HW[3], BDGP[4], CIFAR10, CIFAR100 [29], and YoutubeFace[5]. Table 2 shows the general statistics of these datasets. All the datasets are ordered by category. To construct the shuffled case, we randomly arrange the samples in all datasets.

Competitors: We adopt seven state-of-the-art MVC methods as baselines, including **LVMSC (AAAI 2020)** [20], **AWMVC (AAAI 2023)** [42], **CAMVC (AAAI 2024)** [51],**3AMVC (ACM MM 2024)** [35], **ASR-ETR (ICCV 2023)** [14], **TLSpNM (TPAMI 2023)** [13], and **ARLRR (TPAMI 2023)** [40]. For a fair comparison, we use the official codes of these baselines to conduct the experiments.

[1]https://lig-membres.imag.fr/grimal/data.html
[2]http://mlg.ucd.ie/datasets/segment.html
[3]http://archive.ics.uci.edu/ml/datasets/Multiple+Features
[4]https://www.fruitfly.org/
[5]https://www.cs.tau.ac.il// wolf/ytfaces/

Evaluation Metrics: To measure the clustering performance, five metrics are adopted, including Accuracy (ACC), Normalized Mutual Information (NMI), Purity, the F-score (F), and the Adjusted Rand index (ARI). All the metrics indicate better performance with a larger value. Additionally, the running time is also recorded to evaluate the efficiency.

Parameter Setting: For our proposed SE-FTMC, two parameters need to be tuned, the search range for parameter α is $\{0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 5\}$, and the search range for the number of anchors t is $\{c, 2c, \cdots, 8c\}$. For all the baselines, their related parameters are tuned with the ranges suggested in the original paper and the best results are shown. We experiment with five shuffled cases and give the mean and variance of the results.

5.2 Result Analysis

The experimental results are reported in Table 3 and 4, where the **best** and second best values in all algorithms are indicated by red values and blue values, respectively. From the results, we can draw the following conclusions.

Table 4: Results of our proposed method and other compared methods on CIFAR10-Shuffled, CIFAR100-Shuffled, and YoutubeFace-Shuffled datasets. 'OM' means the "out-of-memory error".

Methods	Complexity Time	Complexity Space	ACC(%)	NMI(%)	PUR(%)	F-score(%)	ARI(%)	Running time(s)
CIFAR10-Shuffled								
LVMSC (AAAI 2020)	$O(n)$	$O(n)$	99.24(0.03)	97.91(0.12)	99.24(0.03)	98.50(0.06)	98.34(0.07)	114.65(10.46)
AWMVC (AAAI 2023)	$O(n)$	$O(n^2)$	91.28(0.84)	85.75(0.65)	91.66(0.98)	86.66(0.83)	85.14(0.93)	1329.05(5.52)
CAMVC (AAAI 2024)	$O(n)$	$O(n)$	99.37(0.21)	98.24(0.57)	99.37(0.21)	98.75(0.41)	98.61(0.45)	79.2(5.28)
3AMVC (ACM MM 2024)	$O(n)$	$O(n)$	98.62(0.55)	97.36(0.01)	98.76(0.35)	97.83(0.36)	97.58(0.41)	442.44(13.57)
ASR-ETR (ICCV 2023)	$O(n\log(n))$	$O(n)$	99.35(0.00)	98.22(0.00)	99.35(0.00)	98.71(0.00)	98.57(0.00)	202.02(10.72)
TLSpNM (TPAMI 2023)	$O(n^2\log(n))$	$O(n^2)$	OM	OM	OM	OM	OM	OM
ARLRR (TPAMI 2023)	$O(n^2\log(n))$	$O(n^2)$	OM	OM	OM	OM	OM	OM
OURS	$O(n)$	$O(n)$	99.42(0.02)	98.36(0.04)	99.42(0.02)	98.85(0.04)	98.72(0.04)	110.47(5.67)
CIFAR100-Shuffled								
LVMSC (AAAI 2020)	$O(n)$	$O(n)$	75.08(3.43)	91.13(2.98)	93.07(0.96)	55.11(14.99)	54.46(15.29)	837.77(16.54)
AWMVC (AAAI 2023)	$O(n)$	$O(n^2)$	86.54(1.08)	97.79(0.18)	90.42(0.74)	88.47(0.99)	88.34(1.00)	1247.83(29.36)
CAMVC (AAAI 2024)	$O(n)$	$O(n)$	90.50(0.27)	98.43(0.04)	93.14(0.25)	91.77(0.28)	91.68(0.28)	268.49(20.31)
3AMVC (ACM MM 2024)	$O(n)$	$O(n)$	88.57(1.14)	97.71(0.47)	91.09(1.12)	88.79(1.82)	88.67(1.84)	732.37(25.41)
ASR-ETR (ICCV 2023)	$O(n\log(n))$	$O(n)$	84.82(0.99)	96.50(0.31)	89.32(0.58)	85.91(1.78)	85.75(1.80)	282.10(37.06)
TLSpNM (TPAMI 2023)	$O(n^2\log(n))$	$O(n^2)$	OM	OM	OM	OM	OM	OM
ARLRR (TPAMI 2023)	$O(n^2\log(n))$	$O(n^2)$	OM	OM	OM	OM	OM	OM
OURS	$O(n)$	$O(n)$	91.94(1.66)	98.18(0.38)	94.31(1.15)	92.63(1.61)	92.55(1.63)	125.20(1.15)
YoutubeFace-Shuffled								
LVMSC (AAAI 2020)	$O(n)$	$O(n)$	14.62(0.49)	11.98(0.45)	20.63(2.06)	8.16(0.71)	1.88(0.33)	401.52(10.32)
AWMVC (AAAI 2023)	$O(n)$	$O(n^2)$	OM	OM	OM	OM	OM	OM
CAMVC (AAAI 2024)	$O(n)$	$O(n)$	15.37(0.26)	13.07(0.26)	29.69(0.39)	7.46(0.10)	2.29(0.08)	529.36(23.45)
3AMVC (ACM MM 2024)	$O(n)$	$O(n)$	17.75(5.90)	16.08(4.29)	30.12(3.31)	7.88(1.37)	3.21(1.41)	1439.72(53.42)
ASR-ETR (ICCV 2023)	$O(n\log(n))$	$O(n)$	14.80(0.44)	11.86(0.47)	28.73(0.45)	7.02(0.03)	2.17(0.14)	376.83(6.98)
TLSpNM (TPAMI 2023)	$O(n^2\log(n))$	$O(n^2)$	OM	OM	OM	OM	OM	OM
ARLRR (TPAMI 2023)	$O(n^2\log(n))$	$O(n^2)$	OM	OM	OM	OM	OM	OM
OURS	$O(n)$	$O(n)$	26.81(1.05)	20.57(1.57)	34.30(1.07)	12.95(0.31)	5.01(0.50)	245.25(20.31)

Table 5: Stability Analysis: Results (ACC%) of tensorial methods on four datasets (ordered case and shuffled case). 'Difference' means the difference between the clustering performance of the ordered case and the shuffled case.

Methods	NGs Ordered	NGs Shuffled	NGs Difference	BBCSport Ordered	BBCSport Shuffled	BBCSport Difference	BDGP Ordered	BDGP Shuffled	BDGP Difference	HW Ordered	HW Shuffled	HW Difference
ASR-ETR (ICCV2023)	100	95.36	4.64	100	83.27	16.73	96.60	52.18	44.42	99.90	82.94	16.69
TLSpNM (TPAMI2023)	100	98.80	1.20	99.82	75.77	24.05	96.76	54.34	42.42	99.95	92.10	7.85
ARLRR (TPAMI2023)	100	97.27	2.73	100	88.85	11.15	60.68	45.56	15.12	99.90	87.68	12.22
OURS	98.80	98.84	0.04	91.91	92.94	1.03	59.08	60.90	1.82	93.30	93.46	0.16

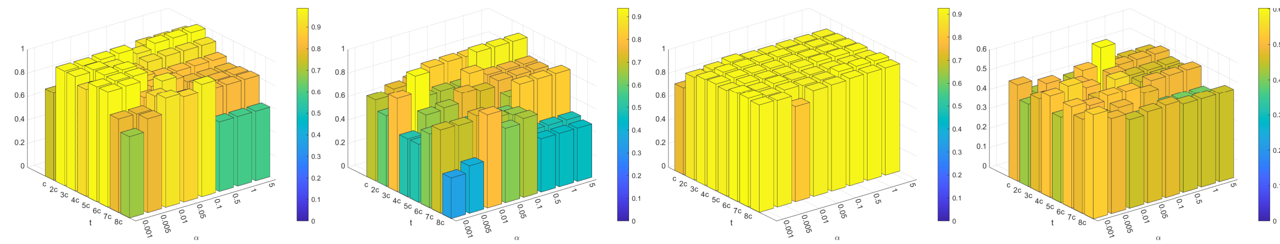
1) Compared to traditional TMC methods (TLSpNM and ARLRR), anchor-based TMC methods like ASR-ETR and our SE-FTMC can handle large-scale datasets. The main reason is that the latter adopts the anchor strategy to reduce the dimension of the representation tensor, thus greatly alleviating the time burden associated with tensor operations.

2) Compared with LVMSC, some methods such as AWMVC, CAMVC, ASR-ETR, and our SE-FTMC, can obtain better clustering results in most cases. This is mainly because the former employs the pre-defined anchor strategy, and the latter dynamically learns optimal anchors to cover the whole data cloud, which avoids the sub-optimal solution problem and improves the discriminability of the learned affinity matrix for clustering.

3) Although our SE-FTMC and ASR-ETR are both anchor-based TMC methods, SE-FTMC shows consistently competitive results on different datasets. For example, in all datasets, SE-FTMC demonstrates shorter running times compared to ASR-ETR. This is because ASR-ETR adopts the sample-oriented low-rank tensor learning framework, which is sensitive to the arrangement of samples and usually takes a time complexity of $O(n\log(n))$. In contrast, SE-FTMC performs the low-rank tensor learning framework in anchor space, which enjoys a time complexity of $O(n)$ and is independent of how the samples are arranged.

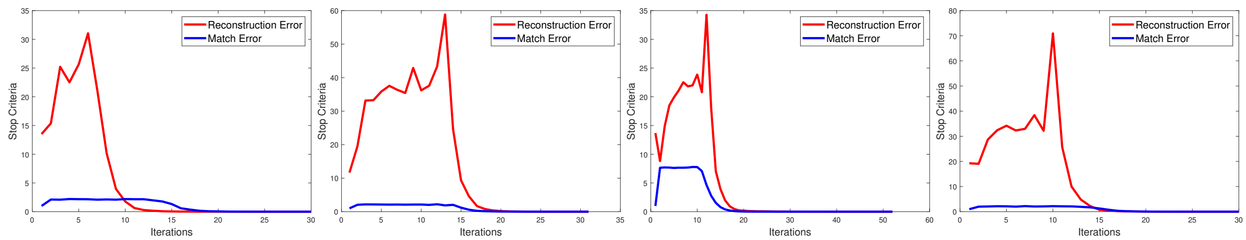
5.3 Stability Analysis

In this section, we analyze the stability of the TMC methods in terms of whether the dataset is shuffled or ordered. Since most tensorial methods have high computational complexity, we conduct experiments on four small-scale datasets, including NGs, BBCSport, BDGP, and HW datasets. We report the clustering performance (ACC) in Table 5. We can observe that sample-oriented TMC methods (e.g.,



(a) NGs (b) BBCSport (c) HW (d) BDGP

Figure 3: The performance (ACC) of our proposed method on four datasets by varying the parameters (α and t).



(a) NGs (b) BBCSport (c) HW (d) BDGP

Figure 4: Convergence curves of our proposed method on four datasets.

ASR-ETR, TLSpNM, and ARLRR) all achieve excellent clustering results when the samples are ordered. However, when the samples are randomly arranged, their performance degrades drastically, especially on the HW and BDGP datasets. This phenomenon suggests that the prior structural information provided by ordered samples can greatly improve sample-oriented TMC methods, but it also highlights their susceptibility to sample arrangement, suggesting that these algorithms may be unstable due to the unknown ordering of samples in real-world datasets. In contrast, our SE-FTMC exhibits stability regardless of the sorting or shuffling of the data, underscoring the effectiveness of our proposed anchor-oriented strategy in enhancing the stability of traditional tensor methods. Note that current TMC methods (TLSpNM, ARLRR, and ASR-ETR) employ different tensor ranks in the low-rank tensor learning framework, all of these methods suffer from a collapse in accuracy when faced with shuffled data. This issue stems from the fact that, regardless of the tensor rank chosen, these methods remain inherently constrained by the sample space, making it challenging for them to effectively handle data variations such as shuffling.

5.4 Model Analysis

Parameter Analysis: The SE-FTMC model has two parameters that need to be tuned, including the trade-off parameter α and the number of anchors t. In detail, the search ranges of α and t are defined as $\{0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 5\}$ and $\{c, 2c, \cdots, 8c\}$, respectively. We record the clustering performance (ACC) of our SE-FTMC on four datasets (NGs, BBCSport, HW, and BDGP) and present the results in Figure. 3. We can observe that our proposed method achieves relatively stable clustering results against these two parameters. From the details, different datasets have varying sensitivities to these parameters. In the HW dataset, the difference between the optimal and lowest values is smaller than that in the BBCSport dataset. Therefore, for other different datasets, we always perform the grid search strategy over the defined search ranges to obtain the optimal clustering results, which is not costly due to the linear complexity of our algorithm and the small search range.

Convergence Analysis: Note that all sub-problems in Eq. (9) are closed and can be solved with guaranteed convergence. Specifically, according to the convergence conditions in Chapter 3 of the ADMM framework [1], the convergence of Algorithm 1 is guaranteed. In this section, we adopt the stopping criteria to further demonstrate the convergence. The stopping criteria used here are Reconstruction Error (RE) and Match Error (ME) [47]. The variation curves of these two stopping criteria on five different datasets are presented in Figure. 4. It is clear to see that the values of RE and ME drop quickly to 0 and remain stable on all datasets, and the number of iterations on these datasets is within 20 steps. Thus, the convergence of the SE-FTMC is not only theoretically established but also experimentally verified.

6 Conclusion

In this paper, we propose an efficient and robust framework, SE-FTMC, for tensorial multi-view clustering. SE-FTMC first discusses the impact of shuffled datasets on TMC methods and reduces the computational complexity of the TMC algorithm to linear time. Unlike existing approaches that apply low-rank tensor learning in the sample space, SE-FTMC leverages this framework in the anchor space to extract high-order correlations among views. A learned correspondence matrix facilitates the transfer of information from anchors to samples for clustering purposes. In addition, we devise an alternative optimization algorithm to solve the proposed model, which enjoys linear complexity. Extensive experiments demonstrate the effectiveness and superiority of our proposed SE-FTMC. Although SE-FTMC has demonstrated satisfactory performance, the leaned anchor sets under orthogonality constraints may have difficulty covering complex sample spaces. Exploring a more stable set of optimal anchors is a promising direction for future research.

Acknowledgments

This work was supported by the Natural Science Foundation of Hebei Province (No. F2025105018), the Tangshan Municipal Science and Technology Plan Project (No.23130225E) and the Beijing Natural Science Foundation (No.4242046).

References

[1] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning* 3, 1 (2011), 1–122.

[2] Guoqing Chao, Shiliang Sun, and Jinbo Bi. 2021. A survey on multiview clustering. *IEEE Transactions on Artificial Intelligence* 2, 2 (2021), 146–168.

[3] Man-Sheng Chen, Chang-Dong Wang, Dong Huang, Jian-Huang Lai, and Philip S Yu. 2022. Efficient orthogonal multi-view subspace clustering. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 127–135.

[4] Man-Sheng Chen, Chang-Dong Wang, and Jian-Huang Lai. 2022. Low-rank tensor based proximity learning for multi-view clustering. *IEEE Transactions on Knowledge and Data Engineering* 35, 5 (2022), 5076–5090.

[5] Yongyong Chen, Shuqin Wang, Xiaolin Xiao, Youfa Liu, Zhongyun Hua, and Yicong Zhou. 2021. Self-paced enhanced low-rank tensor kernelized multi-view subspace clustering. *IEEE Transactions on Multimedia* 24 (2021), 4054–4066.

[6] Jinrong Cui, Yuting Li, Han Huang, and Jie Wen. 2024. Dual contrast-driven deep multi-view clustering. *IEEE Transactions on Image Processing* 33 (2024), 4753–4764.

[7] Fadi Dornaika and Sally El Hajjar. 2024. An End-to-End Approach for Graph-based Multi-view Data Clustering. *IEEE Transactions on Big Data* 10, 5 (2024), 644–654.

[8] Tingting Du, Wei Zheng, and Xingang Xu. 2024. Composite attention mechanism network for deep contrastive multi-view clustering. *Neural Networks* 176 (2024), 106361.

[9] Wei Feng, Dongyuan Wei, Qianqian Wang, Bo Dong, and Quanxue Gao. 2024. Multi-view clustering based on deep non-negative tensor factorization. In *Proceedings of the 32nd ACM International Conference on Multimedia*. 1130–1138.

[10] Lele Fu, Zhaoliang Chen, Yongyong Chen, and Shiping Wang. 2022. Unified low-rank tensor learning and spectral embedding for multi-view subspace clustering. *IEEE Transactions on Multimedia* 25 (2022), 4972–4985.

[11] Zhibin Gu, Zhendong Li, and Songhe Feng. 2024. Topology-Driven Multi-View Clustering via Tensorial Refined Sigmoid Rank Minimization. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 920–931.

[12] Stephan Günnemann, Ines Färber, and Thomas Seidl. 2012. Multi-view clustering using mixture models in subspace projections. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 132–140.

[13] Jipeng Guo, Yanfeng Sun, Junbin Gao, Yongli Hu, and Baocai Yin. 2023. Logarithmic Schatten-p p Norm Minimization for Tensorial Multi-View Subspace Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 3 (2023), 3396–3410.

[14] Jintian Ji and Songhe Feng. 2023. Anchor Structure Regularization Induced Multi-view Subspace Clustering via Enhanced Tensor Rank Minimization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 19343–19352.

[15] Jintian Ji and Songhe Feng. 2023. High-order complementarity induced fast multi-view clustering with enhanced tensor rank minimization. In *Proceedings of the 31st ACM International Conference on Multimedia*. 328–336.

[16] Jintian Ji and Songhe Feng. 2025. Anchors Crash Tensor: Efficient and Scalable Tensorial Multi-view Subspace Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2025), 1–17.

[17] Jintian Ji and Songhe Feng. 2025. Partition-level fusion induced multi-view Subspace Clustering with Tensorial Geman Rank. *Neural Networks* 182 (2025), 106849.

[18] Jintian Ji, Songhe Feng, Jie Huang, Taotao Wei, Xiang Feng, Peiwu Lv, and Bing Li. 2025. Incomplete multi-view clustering via efficient anchor tensor recovery framework. *Neural Networks* (2025), 107652.

[19] Jintian Ji, Songhe Feng, and Yidong Li. 2024. Tensorized unaligned multi-view clustering with multi-scale representation learning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1246–1256.

[20] Zhao Kang, Wangtao Zhou, Zhitong Zhao, Junming Shao, Meng Han, and Zenglin Xu. 2020. Large-scale Multi-view Subspace Clustering in Linear Time. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 4412–4419.

[21] Misha E Kilmer, Karen Braman, Ning Hao, and Randy C Hoover. 2013. Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging. *SIAM J. Matrix Anal. Appl.* 34, 1 (2013), 148–172.

[22] Misha E Kilmer and Carla D Martin. 2011. Factorization strategies for third-order tensors. *Linear Algebra Appl.* 435, 3 (2011), 641–658.

[23] Jing Li, Quanxue Gao, Qianqian Wang, Ming Yang, and Wei Xia. 2024. Orthogonal non-negative tensor factorization based multi-view clustering. *Advances in Neural Information Processing Systems* 36 (2024).

[24] Xingfeng Li, Yinghui Sun, Quansen Sun, and Zhenwen Ren. 2022. Consensus cluster center guided latent multi-kernel clustering. *IEEE Transactions on Circuits and Systems for Video Technology* 33, 6 (2022), 2864–2876.

[25] Xuelong Li, Han Zhang, Rong Wang, and Feiping Nie. 2020. Multiview clustering: A scalable and parameter-free bipartite graph fusion method. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 1 (2020), 330–344.

[26] Zhouchen Lin, Risheng Liu, and Zhixun Su. 2011. Linearized alternating direction method with adaptive penalty for low-rank representation. *Advances in Neural Information Processing Systems* 24 (2011).

[27] Jiyuan Liu, Xinwang Liu, Chuankun Li, Xinhang Wan, Hao Tan, Yi Zhang, Weixuan Liang, Qian Qu, Yu Feng, Renxiang Guan, et al. 2025. Large-scale Multi-view Tensor Clustering with Implicit Linear Kernels. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 20727–20736.

[28] Suyuan Liu, Qing Liao, Siwei Wang, Xinwang Liu, and En Zhu. 2024. Robust and Consistent Anchor Graph Learning for Multi-View Clustering. *IEEE Transactions on Knowledge and Data Engineering* 36, 8 (2024), 4207–4219.

[29] Suyuan Liu, Xinwang Liu, Siwei Wang, Xin Niu, and En Zhu. 2022. Fast Incomplete Multi-view Clustering with View-independent Anchors. *IEEE Transactions on Neural Networks and Learning Systems* (2022).

[30] Suyuan Liu, Siwei Wang, Pei Zhang, Kai Xu, Xinwang Liu, Changwang Zhang, and Feng Gao. 2022. Efficient one-pass multi-view subspace clustering with consensus anchors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 7576–7584.

[31] Xinwang Liu. 2022. Simplemkkm: Simple multiple kernel k-means. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 4 (2022), 5174–5186.

[32] Yongli Liu, Xiaoqin Zhang, Guiying Tang, and Di Wang. 2019. Multi-view subspace clustering based on tensor Schatten-p norm. In *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 5048–5055.

[33] Canyi Lu, Jiashi Feng, Yudong Chen, Wei Liu, Zhouchen Lin, and Shuicheng Yan. 2016. Tensor robust principal component analysis: exact recovery of corrupted low-rank tensors via convex optimization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 5249–5257.

[34] Canyi Lu, Jiashi Feng, Yudong Chen, Wei Liu, Zhouchen Lin, and Shuicheng Yan. 2019. Tensor robust principal component analysis with a new tensor nuclear norm. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42, 4 (2019), 925–938.

[35] Huimin Ma, Siwei Wang, Shengju Yu, Suyuan Liu, Jun-Jie Huang, Huijun Wu, Xinwang Liu, and En Zhu. 2024. Automatic and Aligned Anchor Learning Strategy for Multi-View Clustering. In *Proceedings of the 32nd ACM International Conference on Multimedia*. 5045–5054.

[36] Yalan Qin, Nan Pu, and Hanzhou Wu. 2024. Elastic multi-view subspace clustering with pairwise and high-order correlations. *IEEE Transactions on Knowledge and Data Engineering* 36, 2 (2024), 556–568.

[37] Wei Shen, Yang Yang, and Yinan Liu. 2022. Multi-view clustering for open knowledge base canonicalization. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1578–1588.

[38] Peng Su, Yixi Liu, Shujian Li, Shudong Huang, and Jiancheng Lv. 2024. Robust contrastive multi-view kernel clustering. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*. 4938–4945.

[39] Xiaoli Sun, Rui Zhu, Ming Yang, Xiujun Zhang, and Yuanyan Tang. 2022. Sliced sparse gradient induced multi-view subspace clustering via tensorial arctangent rank minimization. *IEEE Transactions on Knowledge and Data Engineering* 35, 7 (2022), 7483–7496.

[40] Yongqiang Tang, Yuan Xie, and Wensheng Zhang. 2023. Affine subspace robust low-rank self-representation: from matrix to tensor. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 8 (2023), 9357–9373.

[41] Xinhang Wan, Jiyuan Liu, Xinbiao Gan, Xinwang Liu, Siwei Wang, Yi Wen, Tianjiao Wan, and En Zhu. 2024. One-step multi-view clustering with diverse representation. *IEEE Transactions on Neural Networks and Learning Systems* (2024).

[42] Xinhang Wan, Xinwang Liu, Jiyuan Liu, Siwei Wang, Yi Wen, Weixuan Liang, En Zhu, Zhe Liu, and Lu Zhou. 2023. Auto-weighted multi-view clustering for large-scale data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 10078–10086.

[43] Hao Wang, Yan Yang, and Bing Liu. 2019. GMC: Graph-based multi-view clustering. *IEEE Transactions on Knowledge and Data Engineering* 32, 6 (2019), 1116–1129.

[44] Jianlong Wu, Zhouchen Lin, and Hongbin Zha. 2019. Essential tensor learning for multi-view spectral clustering. *IEEE Transactions on Image Processing* 28, 12 (2019), 5910–5922.

[45] Song Wu, Yan Zheng, Yazhou Ren, Jing He, Xiaorong Pu, Shudong Huang, Zhifeng Hao, and Lifang He. 2024. Self-Weighted Contrastive Fusion for Deep Multi-View Clustering. *IEEE Transactions on Multimedia* 26 (2024), 9150–9162.

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

[47] Yuan Xie, Dacheng Tao, Wensheng Zhang, Yan Liu, Lei Zhang, and Yanyun Qu. 2018. On unifying multi-view self-representations for clustering by tensor multi-rank minimization. *International Journal of Computer Vision* 126, 11 (2018), 1157–1179.

[48] Haizhou Yang, Quanxue Gao, Wei Xia, Ming Yang, and Xinbo Gao. 2022. Multiview spectral clustering with bipartite graph. *IEEE Transactions on Image Processing* 31 (2022), 3591–3605.

[49] Xihong Yang, Yue Liu, Sihang Zhou, Siwei Wang, Wenxuan Tu, Qun Zheng, Xinwang Liu, Liming Fang, and En Zhu. 2023. Cluster-guided contrastive graph clustering network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 10834–10842.

[50] Changqing Zhang, Huazhu Fu, Si Liu, Guangcan Liu, and Xiaochun Cao. 2015. Low-rank tensor constrained multiview subspace clustering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1582–1590.

[51] Chao Zhang, Xiuyi Jia, Zechao Li, Chunlin Chen, and Huaxiong Li. 2024. Learning Cluster-Wise Anchors for Multi-View Clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 16696–16704.

[52] Pan Zhou, Canyi Lu, Jiashi Feng, Zhouchen Lin, and Shuicheng Yan. 2019. Tensor low-rank representation for data recovery and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43, 5 (2019), 1718–1732.