Information Fusion 123 (2025) 103322

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Full length article

Mixed-noise robust tensor multi-view clustering via adaptive dictionary learning

Jing-Hua Yang [a], Yi Zhou [a], Lefei Zhang [b], Heng-Chao Li [a,*]

[a] School of Information Science and Technology, Southwest Jiaotong University, China
[b] School of Computer Science, Wuhan University, China

ARTICLE INFO

Keywords:
Mixed-noise robust multi-view clustering
Adaptive dictionary learning
Nonconvex tensor nuclear norm
Alternating direction method of multipliers algorithm

ABSTRACT

Multi-view clustering (MVC) has received extensive attention by exploiting the consistent and complementary information among views. To improve the robustness of MVC, most MVC methods assume that the noise implicit in the data follows a predefined distribution. However, due to equipment limitations and transmission environment, the collected multi-view data often contains mixed noise. The predefined distribution assumption may not be able to effectively suppress complex mixed noise, resulting in a decrease in clustering performance. For solving the above problem, we propose a novel mixed-noise robust tensor multi-view clustering method (MRTMC) via adaptive dictionary learning. To accurately characterize the mixed noise, we consider mixed noise as a combination of structural noise and Gaussian noise and characterize both respectively. Specially, we design adaptive dictionary learning to accurately model structural noise containing semantic information and use Frobenius norm to constrain Gaussian noise. To fully mine the consistency among multiple views, we introduce a nonconvex tensor nuclear norm on the self-representation tensor to explore the high-order correlation among multiple views. Moreover, the weight of each view is learned through an adaptive weighting strategy. For solving the model, we develop an effective algorithm based on the alternating direction method of multipliers (ADMM) framework and provide the convergence guarantee of the algorithm under mild conditions. Extensive experimental results on simulated and real-world datasets indicate the clustering performance of the proposed MRTMC method is superior to the compared methods.

1. Introduction

In reality, a sample is usually represented by multiple features or modalities, which is called multi-view data. For instance, websites can be characterized by text and pictures, and news stories can be reported in several languages. Multi-view data offers more extensive information than single-view data to reveal the inherent structure, and often represents an object more comprehensively [1–6]. In practical application, multi-view data is inevitably polluted by noise due to equipment limitations or transmission environment. Multi-view clustering (MVC) [7–10] aims to mine potential information in multi-view data by exploiting both the consensus and complementary information among multiple views, which has been widely used in fields such as social network analysis, medical diagnosis, and image and video analysis [11–14]. Over the past two decades, numerous MVC methods have been developed and have shown encouraging results. The mainstream MVC methods can generally be classified into two categories: (1) graph learning-based methods; (2) subspace-based methods.

The graph learning-based MVC methods can adaptively learn the similarity graphs from each view and then use spectral clustering on constructed graph to obtain clustering results. Therefore, efficiently utilizing the information from multiple views to construct similarity graphs is crucial for graph learning-based MVC methods [15–24]. For obtaining a better constructed graph, Huang et al. [15] proposed a method similar to multiple kernel learning to identify the optimal combination of similarity graphs. Xia et al. [18] proposed a robust spectral clustering algorithm that recovered a shared transition probability matrix via low-rank and sparse decomposition to obtain a reliable constructed graph, where noise was constrained by ℓ_1 norm. Nie et al. [19] proposed a self-weighted MVC method that can consider the difference of each view to obtain a reliable constructed graph, where noise was constrained by Frobenius norm. The above methods got a reliable similarity graph and achieved impressive clustering results. However, they used a fixed norm constraint on noise, which may not be sufficient to handle complex and mixed noise in multi-view data.

---

[*] Corresponding author.
E-mail addresses: yangjinghua110@126.com (J.-H. Yang), yizhou1224@foxmail.com (Y. Zhou), zhanglefei@whu.edu.cn (L. Zhang), lihengchao_78@163.com (H.-C. Li).

https://doi.org/10.1016/j.inffus.2025.103322
Received 21 January 2025; Received in revised form 20 April 2025; Accepted 9 May 2025

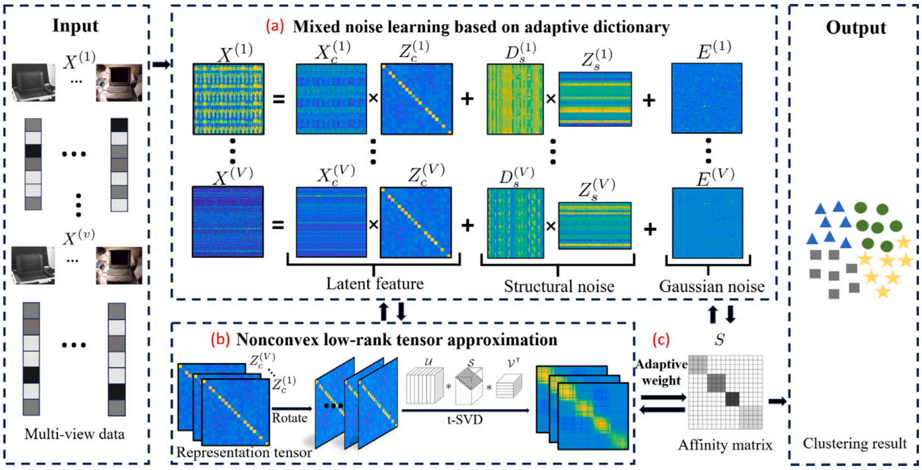


Fig. 1. Framework of MRTMC. Our model consists of three parts, i.e., (a) mixed noise learning based on adaptive dictionary, (b) nonconvex low-rank tensor approximation, and (c) adaptive weighting strategy. Specially, the raw multi-view data is decomposed into latent clean multi-view data, structural noise and Gaussian noise. We use self-representation subspace learning on latent feature to explore the correlation of samples of each view. And the adaptive dictionary learning is designed to describe the view-specific structural noise. Then, the nonconvex low-rank tensor approximation is introduced to characterize the low-rank property of representation tensor $\mathcal{Z}_c$. In addition, the adaptive weighting strategy learns the weight of each view while jointly optimizing similarity matrix S and representation matrix $Z_c^{(v)}$. Finally, the spectral clustering algorithm is applied on the learned affinity matrix to obtain the clustering results.

The subspace-based MVC methods are committed to dividing multi-view data into multiple low-dimensional subspaces. The majority of current multi-view subspace clustering (MVSC) methods typically follow two steps: (1) find the optimal low-dimensional subspace for each view, (2) apply spectral clustering algorithm on affinity matrix which is fused by subspace representations to obtain clustering results. The classic methods of learning affinity matrix are sparse representation and low-rank representation. Base on both, some MVSC methods have been proposed [25–31]. In work [25], Cheng et al. proposed a method which explored low-rank similarity matrices through a joint decomposition of multiple feature matrices. To capture the complementary information among views, Zhang et al. [27] proposed a method which aimed to discover the latent representation of multi-view data. In work [31], Tang et al. proposed a last fusion alignment MVSC method that integrated complementary information at the partition level to reduce the impact of noise. Considering the spatial structure of representation tensor, tensor representation have been widely used in MVSC [32–39]. Tensor-based MVSC methods utilize tensor low-rank representation to exploit the high-order correlation embedded in multiple views. For example, Zhang et al. [32] proposed the tensor unfolding-based method to explore complementary information among multiple views. Different from the work [32], Xie et al. [33] imposed a tensor constraint based on tensor-singular value decomposition (t-SVD) to mine the consensus among multiple views. To explore the application of MVSC in nonlinear space, [34] imposed hyper-Laplacian regularization on subspace coefficient. Chen et al. [35] handled nonlinear data structure through kernel trick. Most of the above models applied t-SVD based tensor nuclear norm (TNN) to approximate representation tensor rank, which is not a reliable approximation of tensor rank. For solving this problem, some methods based on nonconvex tensor rank approximation were proposed [40–46]. In [40,46], the weighted tensor nuclear norm is introduced to explore the high-order correlation of views and the prior information of singular values and achieved promising results. In [44], Pan et al. designed a nonconvex TNN by introducing kernel function to approximate tensor rank. The methods based on nonconvex TNN promote the low-rank property of representation tensor through the nonconvex relaxation, which benefits the affinity matrix.

Most of the above MVSC methods consider the influence of noise on the affinity matrix, and they specified a predefined norm loss in the method, implying that the noise in multi-view data follows a predefined distribution. Specially, $\ell_{2,1}$ norm or ℓ_1 norm is used to handle sample-specific corruptions and outliers [29,32,33,40]. However, the noise in real application is usually mixed and complex, and the noise in multi-view data probably contains semantic features, such as masks and glasses in facial pictures, lighting and angles in object pictures. Therefore, a predefined norm constrain may not be suitable for complex noise situation. To address this problem, we focus on improving the robustness of MVC methods in real-world scene.

In our work, a new mixed-noise robust tensor multi-view clustering method (MRTMC) via adaptive dictionary learning is proposed. The framework is shown in Fig. 1. Firstly, we decompose the raw multi-view data into three parts: latent clean multi-view data, structural noise, and Gaussian noise in Fig. 1(a). Specially, self-representation subspace learning is used on latent clean multi-view data to capture the similarity within each view. Next, we design an adaptive dictionary for each view to learn structural noise which contains semantic information. Meanwhile, Gaussian noise is constrained by Frobenius norm. Secondly, specific-view representation matrices are stacked into a third-order representation tensor. To explore the consistency among views, we introduce nonconvex TNN to characterize the low-rank property of representation tensor, as shown in Fig. 1(b). Subsequently, MRTMC jointly optimizes the representation tensor and affinity matrix through the adaptive weighting strategy, as shown in Fig. 1(c). Finally, the spectral clustering algorithm is applied on the learned affinity matrix to obtain the clustering results. The main contributions of our work are as follows:

(1) We propose a new mixed-noise robust tensor multi-view clustering method (MRTMC) via adaptive dictionary learning, which can not only explore the specific information of each view but also fully mine the consensus information among views. The proposed method can flexibly and effectively solve the mixed noise problem in MVC, which is a challenging task in real-world applications.

(2) To fully mine the consistency among views, we introduce a nonconvex tensor nuclear norm on the self-representation tensor to explore the high-order correlation among multiple views. Moreover, the weight of each view is learned through an adaptive weighting strategy. In addition, mixed noise learning based on adaptive dictionary, nonconvex low-rank tensor approximation, and affinity matrix learning are optimized in a unified framework and promote each other.

(3) An efficient algorithm based on alternating direction method of multipliers (ADMM) framework is developed to solve the proposed nonconvex model. Theoretically, we show the convergence guarantee of the algorithm under mild conditions. Experimental results on simulated and real-world datasets demonstrate the proposed method has better clustering performance than the compared methods.

Table 1
Basic notations and their meanings.

Notation	Meaning
$\mathcal{X}$, X, $\mathbf{x}$, x	Tensor, matrix, vector, and scalar
$\mathcal{X}^{(k)}$	The kth frontal slice of tensor $\mathcal{X}$
$X_{(j)}$	The jth column of matrix X
V, n	The number of views, instances
d_v	Dimension of the vth feature
$\hat{\mathcal{X}} = \mathbf{fft}(\mathcal{X},[],3)$	FFT along tube fiber
$\mathcal{X}^T$, X^T	The transpose of tensor, matrix
$\lVert\cdot\rVert_*$	Matrix nuclear norm
$\lVert\cdot\rVert_{\circledast}$	Tensor norm based on t-SVD
$\lVert\cdot\rVert_1, \lVert\cdot\rVert_F$	ℓ_1 norm, Frobenius norm
$\mathbb{R}$	The real space

The remainder of this paper is structured as follows: Section 2 provides the required notations and preliminaries. Section 3 describes the proposed method in detail. Section 4 presents the experimental results and some discussions, with the conclusion in Section 5.

2. Notations and preliminaries

In this section, some notations and preliminaries needed throughout this paper are introduced. The basic notations and their corresponding meanings are summarized in Table 1.

Before introducing t-SVD, some operators need to be introduced. For a tensor $\mathcal{X} \in \mathbb{R}^{n_1\times n_2\times n_3}$, its block diagonal matrix $\mathbf{bdiag}(\mathcal{X})$ and block circular matrix $\mathbf{bcirc}(\mathcal{X})$ are defined as

$$\mathbf{bdiag}(\mathcal{X}) = \begin{bmatrix} \mathcal{X}^{(1)} & & & \\ & \mathcal{X}^{(2)} & & \\ & & \ddots & \\ & & & \mathcal{X}^{(n_3)} \end{bmatrix},$$

$$\mathbf{bcirc}(\mathcal{X}) = \begin{bmatrix} \mathcal{X}^{(1)} & \mathcal{X}^{(n_3)} & \cdots & \mathcal{X}^{(2)} \\ \mathcal{X}^{(2)} & \mathcal{X}^{(1)} & \cdots & \mathcal{X}^{(3)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{X}^{(n_3)} & \mathcal{X}^{(n_3-1)} & \cdots & \mathcal{X}^{(1)} \end{bmatrix}.$$

The block vectorization is defined as $\mathbf{bvec}(\mathcal{X}) = [\mathcal{X}^{(1)};\cdots;\mathcal{X}^{(n_3)}]$, and the inverse operations of **bdiag** and **bvec** are defined as $\mathbf{bvfold}(\mathbf{bdiag}(\mathcal{X})) = \mathcal{X}$ and $\mathbf{bdfold}(\mathbf{bvec}(\mathcal{X})) = \mathcal{X}$, respectively.

Definition 1 (*t-Product [47]*)**.** For two tensors $\mathcal{X} \in \mathbb{R}^{n_1\times n_2\times n_3}$ and $\mathcal{Y} \in \mathbb{R}^{n_2\times n_4\times n_3}$, the t-product $\mathcal{X} * \mathcal{Y}$ is a tensor of size $n_1 \times n_4 \times n_3$

$$\mathcal{C} = \mathcal{X} * \mathcal{Y} = \mathbf{bvfold}(\mathbf{bcirc}(\mathcal{X})\mathbf{bvec}(\mathcal{Y})).$$

Definition 2 (*f-Diagonal Tensor [47]*)**.** A tensor $\mathcal{X} \in \mathbb{R}^{n_1\times n_2\times n_3}$ is called f-diagonal if all of its frontal slices are diagonal matrices.

Definition 3 (*Orthogonal Tensor [47]*)**.** A tensor $\mathcal{X} \in \mathbb{R}^{n_1\times n_2\times n_3}$ is orthogonal if

$$\mathcal{X}^T * \mathcal{X} = \mathcal{X} * \mathcal{X}^T = \mathcal{I},$$

where $\mathcal{I} \in \mathbb{R}^{n_1\times n_1\times n_3}$ is the identity tensor, with its first frontal slice being the $n \times n$ identify matrix and all other frontal slices being zero.

Definition 4 (*Tensor Singular Value Decomposition (t-SVD) [47]*)**.** A tensor $\mathcal{X} \in \mathbb{R}^{n_1\times n_2\times n_3}$ can be decomposed by t-SVD as

$$\mathcal{X} = \mathcal{U} * \mathcal{S} * \mathcal{V}^T,$$

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. $\mathcal{S} \in \mathbb{R}^{n_1\times n_2\times n_3}$ is a f-diagonal tensor.

Definition 5 (*t-SVD Based Tensor Nuclear Norm (TNN) [48]*)**.** The tensor nuclear norm based on t-SVD of a tensor $\mathcal{X} \in \mathbb{R}^{n_1\times n_2\times n_3}$ is defined as the sum of singular values from all frontal slices of $\hat{\mathcal{X}}$:

$$\lVert\mathcal{X}\rVert_{\circledast} = \sum_{i=1}^{\min\{n_1,n_2\}} \sum_{k=1}^{n_3} |\hat{\mathcal{S}}(i,i,k)|,$$

where $\hat{\mathcal{X}}$ means the fast Fourier transformation (FFT) of $\mathcal{X}$ along tube fiber, $\hat{\mathcal{X}}^{(k)} = \hat{\mathcal{U}}^{(k)}\hat{\mathcal{S}}^{(k)}\hat{\mathcal{V}}^{(k)T}$.

3. The proposed method

In this section, we show the proposed model and present the optimization algorithm. Then, we provide convergence guarantee of the algorithm and analyze the computational complexity.

3.1. Problem formulation

Given a multi-view dataset with n samples and V views $\{X^{(v)}\}_{v=1}^{V}$, $X^{(v)} = [\mathbf{x}_1^{(v)},\ldots,\mathbf{x}_n^{(v)}] \in \mathbb{R}^{d_v\times n}$ denotes the vth view feature, d_v is the dimension of the vth view feature. In real application, the noise in multi-view data is typically mixed and complex, and may contain semantic information. For enhancing the robustness of the MVC methods, the raw multi-view data is decomposed into three parts: latent multi-view clean data $X_c^{(v)}$, structural noise $X_s^{(v)}$, and Gaussian noise $E^{(v)}$ as shown in Fig. 1(a), which can be denoted as

$$X^{(v)} = X_c^{(v)} + X_s^{(v)} + E^{(v)}, v = 1,\ldots,V. \tag{1}$$

Since the latent multi-view clean data is usually distributed in several subspaces, we explore the multiple low-dimensional subspaces to preserve the global structure of the latent clean data. Inspired by the self-representation based subspace clustering, which is based on the idea that samples within a specific subspace can be linearly represented by other samples from the same subspace, latent multi-view clean data can be presented as $X_c^{(v)} = X_c^{(v)}Z_c^{(v)}$. Simultaneously, due to the semantic characteristics of structural noise, we design a structural noise dictionary $D_s^{(v)} \in \mathbb{R}^{d_v\times k_s}$ and use the atoms to adaptively represent the structural noise for each view (k_s is the dictionary atom number), namely, $X_s^{(v)} = D_s^{(v)}Z_s^{(v)}$. Thus, Eq. (1) can be written as follows:

$$X^{(v)} = X_c^{(v)}Z_c^{(v)} + D_s^{(v)}Z_s^{(v)} + E^{(v)}, \tag{2}$$

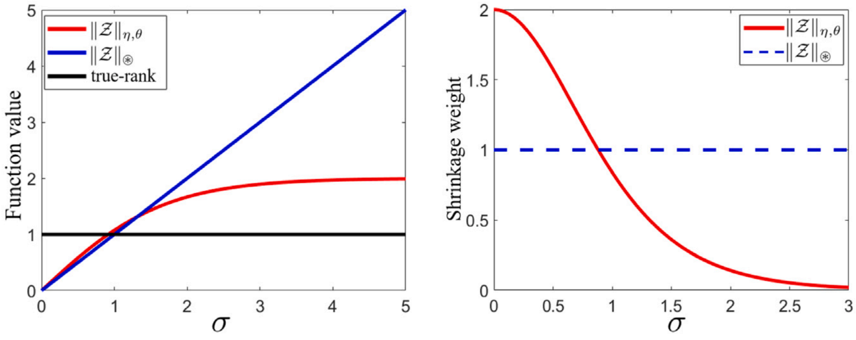
where $X_c^{(v)} \in \mathbb{R}^{d_v\times n}$ is the latent multi-view data, $Z_c^{(v)} \in \mathbb{R}^{n\times n}$ is the representation matrix, $Z_s^{(v)} \in \mathbb{R}^{k_s\times n}$ is the dictionary coefficient matrix.

Since structural noise is usually sparse, we apply ℓ_1 norm to constrain dictionary coefficient matrix $Z_s^{(v)}$ and apply the Frobenius norm to depict Gaussian noise $E^{(v)}$. So the optimization problem can be formulated as:

$$\begin{aligned} &\min_{Z_c^{(v)},D_s^{(v)},Z_s^{(v)}} \sum_{v=1}^{V} \lVert X^{(v)} - X_c^{(v)}Z_c^{(v)} - D_s^{(v)}Z_s^{(v)}\rVert_F^2 + \lVert Z_s^{(v)}\rVert_1, \\ &\text{s.t. } D_s^{(v)} = \left\{ D_s^{(v)} : \lVert D_{s(j)}^{(v)}\rVert_F^2 \leqslant 1, j = 1,2,\ldots,k_s \right\}, v = 1,\ldots,V, \end{aligned} \tag{3}$$

where $D_s^{(v)}$ and $Z_s^{(v)}$ are iteratively optimized with the model, the dictionary of structural noise can be adaptively learned to describe different structural noise of each view. For simplicity, we represent the $D_s^{(v)}$ constraint as $D_s^{(v)} \in \Theta, v = 1,\ldots,V$.

To explore the consistency among the views, some MVSC methods [33–36] introduced t-SVD based TNN (see Definition 5) to mine the high-order correlation among multiple views. However, as a convex approximation of tensor rank, t-SVD based TNN measures the ℓ_1 norm of nonzero singular values, which leads to rank deviation. Moreover, it cannot retain the main information well because of the same shrinkage to each singular value. The large singular value usually represents the main information in the object, so it would be better to give the large



(a) Rank approximation (b) Singular value shrinkage

Fig. 2. The rank approximation and singular value shrinkage of the nonconvex tensor nuclear norm.

singular value a small shrinkage. In work [44], a nonconvex tensor nuclear norm which can flexibly adjust shrinkage of singular value was proposed:

$$\|\mathcal{Z}_c\|_{\eta,\theta} = \sum_j \sum_i \eta * tanh\left(\theta * \sigma_i(\hat{\mathcal{Z}}_c{}^{(j)})\right), \tag{4}$$

where $\hat{\mathcal{Z}}_c = \textbf{fft}\ (\mathcal{Z}_c, [\ ], 3)$, $\sigma_i(\hat{\mathcal{Z}}_c{}^{(j)})$ denotes the ith largest singular value of $\hat{\mathcal{Z}}_c{}^{(j)}$, $\tanh(\mathrm{x}) = (\mathrm{e}^{\mathrm{x}} - \mathrm{e}^{-\mathrm{x}})/(\mathrm{e}^{\mathrm{x}} + \mathrm{e}^{-\mathrm{x}})$. η and θ are two positive parameters to balance shrinkage.

As shown in Fig. 2(a), the nonconvex tensor nuclear norm has a more accurate rank approximation. Besides, each singular value has a more reasonable shrinkage from Fig. 2(b). Inspired by this, we stack the representation matrices $\{Z_c^{(v)}\}_{v=1}^V$ into a third-order representation tensor $\mathcal{Z}_c$ and introduce the nonconvex TNN [44] on $\mathcal{Z}_c$ to characterize its low-rank property. Therefore, Eq. (3) can be formulated as:

$$\begin{aligned}\min_{\mathcal{Z}_c, D_s^{(v)}, Z_s^{(v)}} & \|\mathcal{Z}_c\|_{\eta,\theta} + \sum_{v=1}^V (\|X^{(v)} - X_c^{(v)} Z_c^{(v)} - D_s^{(v)} Z_s^{(v)}\|_F^2 \\ & + \|Z_s^{(v)}\|_1), \\ \text{s.t.}\ & \mathcal{Z}_c = \Phi(Z_c^{(1)}, Z_c^{(2)}, \dots, Z_c^{(V)}), D_s^{(v)} \in \Theta, v = 1, \dots, V,\end{aligned} \tag{5}$$

where $\Phi(\cdot)$ represents the construction of tensor $\mathcal{Z}_c \in \mathbb{R}^{n \times V \times n}$.

Moreover, due to the different contributions of each view to clustering performance, an adaptive weighting strategy is used to distinguish the difference of each view and the affinity matrix and representation tensor are jointly optimized. Finally, our model can be formulated as:

$$\begin{aligned}\min_{\substack{\mathcal{Z}_c, S, w_v \\ D_s^{(v)}, Z_s^{(v)}}} & \lambda_1 \|\mathcal{Z}_c\|_{\eta,\theta} + \sum_{v=1}^V (\|X^{(v)} - X_c^{(v)} Z_c^{(v)} - D_s^{(v)} Z_s^{(v)}\|_F^2 \\ & + \lambda_2 \|Z_s^{(v)}\|_1 + w_v \|S - Z_c^{(v)}\|_F^2), \\ \text{s.t.}\ & \mathcal{Z}_c = \Phi(Z_c^{(1)}, Z_c^{(2)}, \dots, Z_c^{(V)}), D_s^{(v)} \in \Theta, v = 1, \dots, V,\end{aligned} \tag{6}$$

where $\mathcal{Z}_c$ is the representation tensor, $S \in \mathbb{R}^{n \times n}$ is the affinity matrix and w_v is the weight of vth view. The parameters λ_1 and λ_2 are used to balance the contributions of all terms in the Eq. (6). $X_c^{(v)}$ and $D_s^{(v)}$ respectively denote latent clean data and the structural noise dictionary, and $Z_c^{(v)}$ and $Z_s^{(v)}$ are two corresponding coefficient matrices.

Our model consists of four terms. The first term is adopted to fully explore the consistency among views by depicting low-rank property of representation tensor $\mathcal{Z}_c$. The second and third terms denote the mixed noise learning by adaptive dictionary for characterizing the view-specific information. The last term can obtain the unified affinity matrix by adaptive weighting strategy.

Remark 1. The robust weighted low-rank tensor approximation (RWLTA) [41] method also handles the mixed noise problem and introduces three norms ($l_{Cauchy,1}$, l_1, l_F) to respectively constrain outliers, random corruptions, and slight perturbations, then achieves the promising results. Compared with the method RWLTA, the main differences of the proposed method are as follows: First, the proposed method applies the adaptive dictionary learning strategy to flexibly adjust to varying noise characteristics across different datasets, therefore resulting in the better clustering performance. Second, in addition to the numerical convergence of the algorithm, we theoretically prove the convergence guarantee of the proposed algorithm under mild conditions.

3.2. Optimization algorithm

To better apply the optimization framework to solve the proposed model, we rewrite the proposed model as follows:

$$\begin{aligned}\min_{\substack{\mathcal{Z}_c, S, w_v \\ D_s^{(v)}, Z_s^{(v)}}} & \lambda_1 \|\mathcal{Z}_c\|_{\eta,\theta} + \sum_{v=1}^V \big(\ \|X^{(v)} - X_c^{(v)} Z_c^{(v)} - D_s^{(v)} Z_s^{(v)}\|_F^2 \\ & + \lambda_2 \|Z_s^{(v)}\|_1 + w_v \|S - Z_c^{(v)}\|_F^2 + \Psi(D_s^{(v)})\ \big), \\ \text{s.t.}\ & \mathcal{Z}_c = \Phi(Z_c^{(1)}, Z_c^{(2)}, \dots, Z_c^{(V)}),\end{aligned} \tag{7}$$

where $\Psi(D_s^{(v)})$ is indicator function defined as (q is a large value):

$$\Psi(D_s^{(v)}) = \begin{cases} 0, & \text{if } \|D_{s(j)}^{(v)}\|_F^2 \leqslant 1, \\ q, & \text{otherwise.} \end{cases} \tag{8}$$

The solution to Eq. (7) is difficult because objective function is not joint convex. Following the idea of ADMM, we introduce three auxiliary variables to decouple this equation, we define $\mathcal{G} = \mathcal{Z}_c, Y^{(v)} = Z_s^{(v)}, B^{(v)} = D_s^{(v)}, v = 1, \dots, V$. Eq. (7) can be reformulated as:

$$\begin{aligned}\min_{\substack{\mathcal{G}, \mathcal{Z}_c, S, w_v \\ \mathcal{D}_s, \mathcal{Z}_s, \mathcal{Y}, \mathcal{B}}} & \lambda_1 \|\mathcal{G}\|_{\eta,\theta} + \sum_{v=1}^V \big(\ \|X^{(v)} - X_c^{(v)} Z_c^{(v)} - D_s^{(v)} Z_s^{(v)}\|_F^2 \\ & + \lambda_2 \|Y^{(v)}\|_1 + w_v \|S - Z_c^{(v)}\|_F^2 + \Psi(B^{(v)})\ \big), \\ \text{s.t.}\ & \mathcal{Z}_c = \Phi(Z_c^{(1)}, Z_c^{(2)}, \dots, Z_c^{(V)}), \\ & \mathcal{G} = \mathcal{Z}_c, Y^{(v)} = Z_s^{(v)}, B^{(v)} = D_s^{(v)}, v = 1, \dots, V.\end{aligned} \tag{9}$$

The augmented Lagrangian function of Eq. (9) is:

$$\begin{aligned}& \mathcal{L}_\rho(\mathcal{Z}_c, \mathcal{G}, \mathcal{Z}_s, \mathcal{Y}, \mathcal{D}_s, \mathcal{B}, S, w_v; \mathcal{W}, \mathcal{M}, \mathcal{P}) \\ & = \lambda_1 \|\mathcal{G}\|_{\eta,\theta} + \sum_{v=1}^V \big(\ \|X^{(v)} - X_c{}^{(v)} Z_c^{(v)} - D_s^{(v)} Z_s^{(v)}\|_F^2 \\ & + \lambda_2 \|Y^{(v)}\|_1 + w_v \|S - Z_c^{(v)}\|_F^2 + \Psi(B^{(v)})\ \big) \\ & + \langle \mathcal{W}, \mathcal{Z}_c - \mathcal{G} \rangle + \frac{\rho}{2} \|\mathcal{Z}_c - \mathcal{G}\|_F^2 + \langle \mathcal{P}, \mathcal{D}_s - \mathcal{B} \rangle \\ & + \frac{\rho}{2} \|\mathcal{D}_s - \mathcal{B}\|_F^2 + \langle \mathcal{M}, \mathcal{Z}_s - \mathcal{Y} \rangle + \frac{\rho}{2} \|\mathcal{Z}_s - \mathcal{Y}\|_F^2,\end{aligned} \tag{10}$$

where $\mathcal{W}$, $\mathcal{M}$, and $\mathcal{P}$ are the Lagrangian multipliers, ρ is the penalty parameter, $\langle \cdot, \cdot \rangle$ is the inner product. Within the ADMM framework [49–51], we can update the remaining variable by keeping the other variables fixed at their latest values. The overall iterative procedure of the algorithm is as follows:

$$\begin{cases}
\mathcal{Z}_c^{k+1} = \underset{\mathcal{Z}_c}{argmin}\ \mathcal{L}_\rho(\mathcal{Z}_c^k, \mathcal{G}^k, \mathcal{Z}_s^k, \mathcal{Y}^k, \mathcal{D}_s^k, \mathcal{B}^k, S^k, w_v^k; \mathcal{W}^k, \mathcal{M}^k, \mathcal{P}^k), \\
\mathcal{G}^{k+1} = \underset{\mathcal{G}}{argmin}\ \mathcal{L}_\rho(\mathcal{Z}_c^{k+1}, \mathcal{G}^k, \mathcal{Z}_s^k, \mathcal{Y}^k, \mathcal{D}_s^k, \mathcal{B}^k, S^k, w_v^k; \mathcal{W}^k, \mathcal{M}^k, \mathcal{P}^k), \\
\mathcal{Z}_s^{k+1} = \underset{\mathcal{Z}_s}{argmin}\ \mathcal{L}_\rho(\mathcal{Z}_c^{k+1}, \mathcal{G}^{k+1}, \mathcal{Z}_s^k, \mathcal{Y}^k, \mathcal{D}_s^k, \mathcal{B}^k, S^k, w_v^k; \mathcal{W}^k, \mathcal{M}^k, \mathcal{P}^k), \\
\mathcal{Y}^{k+1} = \underset{\mathcal{Y}}{argmin}\ \mathcal{L}_\rho(\mathcal{Z}_c^{k+1}, \mathcal{G}^{k+1}, \mathcal{Z}_s^{k+1}, \mathcal{Y}^k, \mathcal{D}_s^k, \mathcal{B}^k, S^k, w_v^k; \mathcal{W}^k, \mathcal{M}^k, \mathcal{P}^k), \\
\mathcal{D}_s^{k+1} = \underset{\mathcal{D}_s}{argmin}\ \mathcal{L}_\rho(\mathcal{Z}_c^{k+1}, \mathcal{G}^{k+1}, \mathcal{Z}_s^{k+1}, \mathcal{Y}^{k+1}, \mathcal{D}_s^k, \mathcal{B}^k, S^k, w_v^k; \mathcal{W}^k, \mathcal{M}^k, \mathcal{P}^k), \\
\mathcal{B}^{k+1} = \underset{\mathcal{B}}{argmin}\ \mathcal{L}_\rho(\mathcal{Z}_c^{k+1}, \mathcal{G}^{k+1}, \mathcal{Z}_s^{k+1}, \mathcal{Y}^{k+1}, \mathcal{D}_s^{k+1}, \mathcal{B}^k, S^k, w_v^k; \mathcal{W}^k, \mathcal{M}^k, \mathcal{P}^k), \\
S^{k+1} = \underset{S}{argmin}\ \mathcal{L}_\rho(\mathcal{Z}_c^{k+1}, \mathcal{G}^{k+1}, \mathcal{Z}_s^{k+1}, \mathcal{Y}^{k+1}, \mathcal{D}_s^{k+1}, \mathcal{B}^{k+1}, S^k, w_v^k; \mathcal{W}^k, \mathcal{M}^k, \mathcal{P}^k), \\
w_v^{k+1} = \underset{w_v}{argmin}\ \mathcal{L}_\rho(\mathcal{Z}_c^{k+1}, \mathcal{G}^{k+1}, \mathcal{Z}_s^{k+1}, \mathcal{Y}^{k+1}, \mathcal{D}_s^{k+1}, \mathcal{B}^{k+1}, S^{k+1}, w_v^k; \mathcal{W}^k, \mathcal{M}^k, \mathcal{P}^k), \\
\mathcal{W}^{k+1} = \mathcal{W}^k + \rho^k(\mathcal{Z}_c^{k+1} - \mathcal{G}^{k+1}), \\
\mathcal{M}^{k+1} = \mathcal{M}^k + \rho^k(\mathcal{Z}_s^{k+1} - \mathcal{Y}^{k+1}), \\
\mathcal{P}^{k+1} = \mathcal{P}^k + \rho^k(\mathcal{D}_s^{k+1} - \mathcal{B}^{k+1}).
\end{cases} \tag{11}$$

The above sub-problems are solved alternately and iteratively until the algorithm converges. Next, we show the details of each subproblem.

Step 1: Updating of $\mathcal{Z}_c$. With other variables fixed, $\mathcal{Z}_c$ can be updated by:

$$\begin{aligned}\min_{\mathcal{Z}_c} \sum_{v=1}^{V} \Big(& \|X^{(v)} - X_c^{(v)} Z_c^{(v)} - D_s^{(v)k} Z_s^{(v)k}\|_F^2 \\ & + w_v^k \|S^k - Z_c^{(v)}\|_F^2 \Big) + \frac{\rho^k}{2}\|\mathcal{Z}_c^k - \mathcal{G}^k + \frac{1}{\rho}\mathcal{W}^k\|_F^2.\end{aligned} \tag{12}$$

Since the updating for each $Z_c^{(v)}$ is independent, we can separate Eq. (12) into V sub-problems. By setting the derivative of each sub-problem to zero, we can derive the closed-form solution for $Z_c^{(v)}$:

$$\begin{aligned}Z_c^{(v)k+1} = & \big((\rho^k + 2w_v^k)I + 2X_c^{(v)^T} X_c^{(v)}\big)^{-1} \\ & \big(\rho^k G^{(v)k} - W^{(v)k} + 2X_c^{(v)^T} H^{(v)k} + 2w_v^k S^k\big),\end{aligned} \tag{13}$$

where I is identity matrix and $H^{(v)k} = X^{(v)} - D_s^{(v)k} Z_s^{(v)k}$.

Step 2: Updating of $\mathcal{G}$. With other variables fixed, Eq. (10) becomes:

$$\min_{\mathcal{G}} \lambda_1 \|\mathcal{G}\|_{\eta,\theta} + \frac{\rho^k}{2}\|\mathcal{G} - (\mathcal{Z}_c^{k+1} + \frac{1}{\rho^k}\mathcal{W}^k)\|_F^2, \tag{14}$$

It is challenging to directly achieve the closed-form solution for above sub-problem. We apply Fourier transform to convert Eq. (14) into frequency domain [45] and split it into n sub-problems, the jth problem is

$$\begin{aligned}\mathcal{G}^{(j)k+1} = & \arg\min_{\hat{\mathcal{G}}^{(j)}} \frac{\lambda_1}{\rho^k} \sum_{i=1}^{V} \Omega\left(\sigma_i\left(\hat{\mathcal{G}}^{(j)}\right), \theta\right) \\ & + \frac{1}{2}\|\hat{\mathcal{G}}^{(j)} - \hat{\mathcal{F}}^{(j)k}\|_F^2,\end{aligned} \tag{15}$$

where $\hat{\mathcal{G}} = \mathbf{fft}\ (\mathcal{G}, [\ \ ], 3)$ and $\hat{\mathcal{F}^k} = \mathbf{fft}(\mathcal{Z}_c^{k+1} + \frac{1}{\rho^k}\mathcal{W}^k, [\ \ ], 3)$. $\hat{\mathcal{G}}^{(j)}$ is the jth frontal slice of $\hat{\mathcal{G}}$, $\Omega(x, \theta) = \eta\, tanh(\theta * x)$.

Based on the antimonotone characteristic of gradient of the non-convex function in Eq. (4) and the definition of the supergradient for concave functions, we obtain:

$$0 \le \nabla\Omega(\sigma_1^k, \theta) \le \nabla\Omega(\sigma_2^k, \theta) \le \cdots \le \nabla\Omega(\sigma_V^k, \theta), \tag{16}$$

$$\Omega(\sigma_i(\hat{\mathcal{G}}^{(j)}), \theta) \le \Omega(\sigma_i^k, \theta) + \nabla\Omega(\sigma_i^k, \theta)(\sigma_i(\hat{\mathcal{G}}^{(j)}) - \sigma_i^k), \tag{17}$$

where σ_i^k denotes the ith singular value of $\hat{\mathcal{G}}^{(j)k}$, $\sigma_1^k \ge \sigma_2^k \ge \cdots \ge \sigma_V^k$, $\nabla\Omega(\sigma_i^k, \theta)$ is the gradient of $\Omega(\sigma_i(\hat{\mathcal{G}}^{(j)}), \theta)$ at σ_i^k. Based on Eq. (17), Eq. (15) can be relaxed into:

$$\begin{aligned}\mathcal{G}^{(j)k+1} & = \arg\min_{\hat{\mathcal{G}}^{(j)}} \frac{\lambda_1}{\rho^k} \sum_{i=1}^{V} \phi(\sigma_i^k, \theta) \\ & + \nabla\Omega(\sigma_i^k, \theta)(\sigma_i(\hat{\mathcal{G}}^{(j)}) - \sigma_i^k) + \frac{1}{2}\|\hat{\mathcal{G}}^{(j)} - \hat{\mathcal{F}}^{(j)k}\|_F^2, \\ & = \arg\min_{\hat{\mathcal{G}}^{(j)}} \frac{\lambda_1}{\rho^k} \sum_{i=1}^{V} \nabla\Omega(\sigma_i^k, \theta)\sigma_i(\hat{\mathcal{G}}^{(j)}) + \frac{1}{2}\|\hat{\mathcal{G}}^{(j)} - \hat{\mathcal{F}}^{(j)k}\|_F^2.\end{aligned} \tag{18}$$

Then, we use the generalized weighted singular value threshold from [52,53] to solve Eq. (18), the optimal solution for Eq. (18) is:

$$\hat{\mathcal{G}}^{(j)k+1} = U \mathcal{J}_{\frac{\lambda_1}{\rho^k}\nabla\Omega}(\Sigma) V^T, \tag{19}$$

where $\mathcal{J}_{\frac{\lambda_1}{\rho^k}\nabla\Omega}(\Sigma) = \operatorname{diag}\left\{\max\left(\Sigma_{ii} - \frac{\lambda_1 \nabla\Omega(\sigma_i^k)}{\rho^k}, 0\right)\right\}$ and the singular value decomposition (SVD) of $\hat{\mathcal{F}}^{(j)k}$ is denoted as $\hat{\mathcal{F}}^{(j)k} = U\Sigma V^T$.

Step 3: Updating of $\mathcal{Z}_s$. With other variables fixed, $\mathcal{Z}_s$ can be updated by:

$$\begin{aligned}\min_{\mathcal{Z}_s} & \sum_{v=1}^{V} \|X^{(v)} - X_c^{(v)} Z_c^{(v)k+1} - D_s^{(v)k} Z_s^{(v)}\|_F^2 \\ & + \frac{\rho^k}{2}\|Z_s^{(v)} - Y^{(v)k} + \frac{1}{\rho^k} M^{(v)k}\|_F^2.\end{aligned} \tag{20}$$

Similar to Eq. (12), we divide Eq. (20) into V sub-problems and set the derivative of each sub-problem to zero, the closed-form solution of $Z_s^{(v)}$can be derived:

$$\begin{aligned}Z_s^{(v)k+1} = & (\rho^k I + 2D_s^{(v)k^T} D_s^{(v)k})^{-1} \\ & \big(\rho^k Y^{(v)k} - M^{(v)k} + 2D_s^{(v)k^T} C^{(v)k}\big),\end{aligned} \tag{21}$$

where $C^{(v)k} = X^{(v)} - X_c^{(v)} Z_c^{(v)k+1}$.

Step 4: Updating of $\mathcal{Y}$. With other variables fixed, $\mathcal{Y}$ can be updated by:

$$\min_{Y^{(v)}} \sum_{v=1}^{V} \lambda_2 \|Y^{(v)}\|_1 + \frac{\rho^k}{2}\|Y^{(v)} - (Z_s^{(v)k+1} + \frac{1}{\rho^k} M^{(v)k})\|_F^2. \tag{22}$$

Similar to Eq. (12), updating of $Y^{(v)}$ for each view is independent and has the following closed-form solution applying the soft shrinkage operator:

$$Y^{(v)k+1} = \max(|A^{(v)k}| - \frac{\lambda_2}{\rho^k}, 0) \circ \frac{A^{(v)k}}{|A^{(v)k}|}, \tag{23}$$

where $A^{(v)k} = Z_s^{(v)k+1} + \frac{1}{\rho^k} M^{(v)k}$, $\circ$ presents component-wise multiplication and a convention is assumed: $0 \circ \frac{0}{0} = 0$.

Step 5: Updating of $\mathcal{D}_s$. With other variables fixed, $\mathcal{D}_s$ can be updated by:

$$\begin{aligned}\min_{\mathcal{D}_s} & \sum_{v=1}^{V} \|X^{(v)} - X_c^{(v)} Z_c^{(v)k+1} - D_s^{(v)} Z_s^{(v)k+1}\|_F^2 \\ & + \frac{\rho^k}{2}\|D_s^{(v)} - B^{(v)k} + \frac{1}{\rho^k} P^{(v)}\|_F^2.\end{aligned} \tag{24}$$

Similar to Eq. (12), updating of $D_s^{(v)}$ for each view is independent and has the following closed-form solution by derivation:

$$\begin{aligned}D_s^{(v)k+1} = & (\rho^k B^{(v)k} - P^{(v)k} + 2C^{(v)k} Z_s^{(v)k+1^T}) \\ & (\rho^k I + 2Z_s^{(v)k+1} Z_s^{(v)k+1^T})^{-1}.\end{aligned} \tag{25}$$

Step 6: Updating of $\mathcal{B}$. With other variables fixed, $\mathcal{B}$ can be updated by:

$$\min_{\mathcal{B}} \sum_{v=1}^{V} \Psi(B^{(v)}) + \frac{\rho^k}{2}\|B^{(v)} - D_s^{(v)k+1} - \frac{1}{\rho^k} P^{(v)k}\|_F^2, \tag{26}$$

where updating of $B^{(v)}$ for each view is independent and has the closed-form solution:

$$B_{(j_1)}^{(v)k+1} = \frac{Q_{(j_1)}^k}{\max\left\{1, \|Q_{(j_1)}^k\|_2^2\right\}}, j_1 = 1, 2, \ldots, k_s, \tag{27}$$

Algorithm 1 Mixed-Noise Robust Tensor Multi-View Clustering via Adaptive Dictionary Learning.

```
Input: multi-view data $X^{(v)}$; parameters: $\lambda_1, \lambda_2, \eta, \theta, \mu, \rho_{max}$ and $k_s$.
Initialize: $\rho = 10^{-3}$, $tol = 10^{-6}$, $\mathcal{G}$, $S$, $Z_c^{(v)}$, $D_s^{(v)}$, $Z_s^{(v)}$, $Y^{(v)}$, $B^{(v)}$, $W^{(v)}$, $M^{(v)}$, $P^{(v)}$, $w_v$, $v = 1, \dots, V$.
while not converged do
    Update $\{Z_c^{(v)}\}_{v=1}^V$ by Eq. (13);
    Update $\mathcal{G}$ by Eq. (19);
    Update $\{Z_s^{(v)}\}_{v=1}^V$ by Eq. (21);
    Update $\{Y^{(v)}\}_{v=1}^V$ by Eq. (23);
    Update $\{D_s^{(v)}\}_{v=1}^V$ by Eq. (25);
    Update $\{B^{(v)}\}_{v=1}^V$ by Eq. (27);
    Update $\{w_v\}_{v=1}^V$ by $w_v = 1/(2\|S - Z_c^{(v)}\|_F)$;
    Update $S$ by Eq. (29);
    Update $\mathcal{W}, \mathcal{M}, \mathcal{P}$ by Eq. (30);
    Update $\rho$ by Eq. (31);
    Check the convergence condition:
        $\max\left\{\begin{array}{l} \|\mathcal{Z}_s^{k+1} - \mathcal{Y}^{k+1}\|_\infty, \\ \|\mathcal{D}_s^{k+1} - \mathcal{B}^{k+1}\|_\infty, \\ \|\mathcal{Z}_c^{k+1} - \mathcal{G}^{k+1}\|_\infty. \end{array}\right\} \leq tol.$
end while
Output: Output matrix $S^{k+1} \in \mathbb{R}^{n \times n}$ and perform spectral clustering on $S^{k+1}$.
```

where $Q_{(j_1)}^k = D_{s(j_1)}^{(v)k+1} + P_{(j_1)}^{(v)k}/\rho^k$.

Step 7: Updating of S. With other variables fixed, Eq. (9) simplifies to:

$$\min_S \sum_v^V w_v \|S - Z_c^{(v)k+1}\|_F^2. \tag{28}$$

Inspired by the work of [19], adaptive weight w_v can be updated by[1] $w_v^{k+1} = 1/(2\|S^k - Z_c^{(v)k+1}\|_F)$. If the vth view significantly impacts the clustering effect, then $\|S - Z_c^{(v)}\|_F$ should be small, so w_v will be large, that is, the view is given a larger weight. S can be updated as follows:

$$S^{k+1} = \left(\sum_{v=1}^V w_v^{k+1} Z_c^{(v)k+1}\right) \Big/ \sum_{v=1}^V w_v^{k+1}. \tag{29}$$

Furthermore, the Lagrange multipliers $\mathcal{W}$, $\mathcal{M}$ and $\mathcal{P}$ are updated by following rules:

$$\begin{aligned} \mathcal{W}^{k+1} &= \mathcal{W}^k + \rho^k(\mathcal{Z}_c^{k+1} - \mathcal{G}^{k+1}), \\ \mathcal{M}^{k+1} &= \mathcal{M}^k + \rho^k(\mathcal{Z}_s^{k+1} - \mathcal{Y}^{k+1}), \\ \mathcal{P}^{k+1} &= \mathcal{P}^k + \rho^k(\mathcal{D}_s^{k+1} - \mathcal{B}^{k+1}). \end{aligned} \tag{30}$$

Finally, the penalty parameter ρ is changed as follows:

$$\rho^{k+1} = min(\mu\rho^k, \rho_{max}), \tag{31}$$

where μ is a positive parameter used to enhance the convergence rate.

In summary, Algorithm 1 shows the pseudocode of the proposed algorithm.

3.3. Convergence analysis

Due to the non-convexity of the proposed model, the theoretical convergence remains a challenge. It is difficult to guarantee the global optimal solution of the proposed Algorithm 1. Theorem 1 is provided as follows, which demonstrates the proposed Algorithm 1 converges to the stationary point.

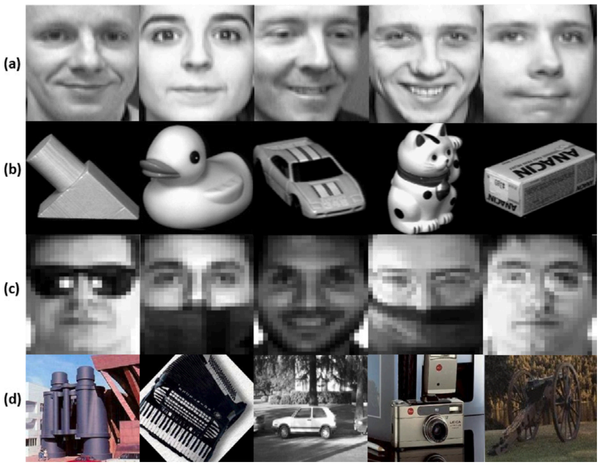


Fig. 3. Some examples of the test datasets: (a) ORL, (b) COIL-20, (c) AR, and (d) Caltech-20.

Theorem 1. *The sequence* $\{\mathcal{Z}_c^k, \mathcal{G}^k, \mathcal{Z}_s^k, \mathcal{Y}^k, \mathcal{D}_s^k, \mathcal{B}^k, S^k, w_v^k, \mathcal{W}^k, \mathcal{M}^k, \mathcal{P}^k\}$ *by Algorithm 1 has at least one accumulation point* $\{\mathcal{Z}_c^*, \mathcal{G}^*, \mathcal{Z}_s^*, \mathcal{Y}^*, \mathcal{D}_s^*, \mathcal{B}^*, S^*, w_v^*, \mathcal{W}^*, \mathcal{M}^*, \mathcal{P}^*\}$. *Then,* $\{\mathcal{Z}_c^*, \mathcal{G}^*, \mathcal{Z}_s^*, \mathcal{Y}^*, \mathcal{D}_s^*, \mathcal{B}^*, S^*, w_v^*\}$ *is the stationary point of Eq. (7) as long as* $\lim_{k\to\infty} \rho^k(\mathcal{G}^{k+1} - \mathcal{G}^k) = 0$, $\lim_{k\to\infty} \rho^k(\mathcal{Y}^{k+1} - \mathcal{Y}^k) = 0$, $\lim_{k\to\infty} \rho^k(\mathcal{B}^{k+1} - \mathcal{B}^k) = 0$, *and* $\sum_{j=1}^{\infty}(\rho^j + \rho^{j-1})/(\rho^{j-1})^2 < \infty$.

For the detailed proof, please refer to the supplementary material.

3.4. Complexity analysis

The main factors affecting the computational complexity of Algorithm 1 are the update of $\mathcal{G}$, $\mathcal{Z}_c$, $\mathcal{D}_s$ and $\mathcal{Z}_s$. The update of $\mathcal{G}$ requires the calculation of fast Fourier transform, inverse Fourier transform and singular value decomposition, which will cost $\mathcal{O}(Vn^2log(n) + V^2n^2)$. The update of $D_s^{(v)}$ requires inversion of $(\rho I + 2Z_s^{(v)}Z_s^{(v)^T})$, while the update of $Z_s^{(v)}$ requires inversion of $(\rho I + 2D_s^{(v)^T}D_s^{(v)})$, both costing $\mathcal{O}\big(Vk_s^3\big)$. Finally, the update of $Z_c^{(v)}$ requires inversion of $\big((\rho + 2w_v)I + 2X_c^{(v)^T}X_c^{(v)}\big)$, costing $\mathcal{O}(Vn^3)$. So the overall computational complexity is: $\mathcal{O}\Big(tV(n^2log(n) + Vn^2 + 2k_s^3 + n^3)\Big)$, where t represents the number of iterations.

4. Experiments

In this section, we evaluate the clustering performance of MRTMC through the simulated experiments and real-world experiments.

4.1. Experimental settings

The experiment selects nine classical datasets for MVC tasks, some example of the datasets are shown in Fig. 3. The detailed information of the datasets are introduced as Table 2, including their sample size, multi-view features and number of features, and number of clusters. In the experiments, the test data are normalized to the range of $[0, 1]$. All experiments are implemented on the platform of Windows 11 and MATLAB R2022b with an Intel(R) Core(TM) i9-13980HX and 32 GB RAM.

Dataset descriptions:MSRC-V1[2]**:** It consists of 210 object pictures, each image has 5 types of features and can be divided into 7 categories.

---

[1] To avoid the denominator being zero, in practice we perform the following operation: $w_v = 1/(2\|S - Z_c^{(v)}\|_F + \delta)$, where δ is a very small value.

[2] https://www.microsoft.com/en-us/msrc.

Table 2
Detailed information of all multi-view datasets.

Datasets	Samples	Clusters	View1	View2	View3	View4	View5	View6
MSRC-V1	210	7	CM(24)	HOG(576)	GIST(512)	LBP(256)	GENTRIST(254)	–
ORL	400	40	Intensity(4096)	LBP(3304)	Gabor(6750)	–	–	–
Yale	165	15	Intensity(4096)	LBP(3304)	Gabor(6750)	–	–	–
COIL-20	1440	20	Intensity(1024)	LBP(4096)	–	–	–	–
AR	1300	100	Intensity(792)	LBP(472)	Gabor(792)	–	–	–
FERET	1400	200	Intensity(1024)	LBP(708)	Gabor(1024)	–	–	–
ALOI	1080	10	Color Similarity(77)	Haralick(13)	HSV(64)	RGB(125)	–	–
Caltech-20	2386	20	Gabor(48)	Wavelet Monments(40)	CENTRIST(254)	HOG(1984)	GIST(512)	LBP(928)
Scene-15	4485	15	PHOG(20)	GIST(59)	LBP(40)	–	–	–

Table 3
The settings of parameters (λ_1, λ_2) for all datasets.

Datasets	MSRC-V1	ORL	Yale	COIL-20	AR	ALOI	FERET	Caltech-20	Scene-15
	λ_1, λ_2	λ_1, λ_2	λ_1, λ_2	λ_1, λ_2	λ_1, λ_2	λ_1, λ_2	λ_1, λ_2	λ_1, λ_2	λ_1, λ_2
$Case1$	10,10	5,5	5,50	50,50	10,50	40,10	10,20	40,50	50,10
$Case2$	10,30	5,20	10,1	40,40					
$Case3$	10,10	5,10	20,10	50,30					
$Case4$	30,10	5,1	5,40	50,1					

ORL[3]**:** The dataset consists of 400 face images with variations in illumination, facial expressions, and details. These images have three types of features and can be divided into 40 categories. **Yale**[4]**:** It consists of 165 face images under different facial expressions and configurations. **COIL-20**[5]**:** It consists of 1440 object images under different angles. These images have two view features and can be divided into 10 categories. **AR:** The dataset comes from [54] and consists of 1300 face images. The illumination and facial expressions of these images are different and some of these images are severely obstructed. **FERET:** The dataset comes from [55] and consists of 1400 face images. These images have different expressions, lighting, posture and age changes. For the above two datasets, the LBP features are obtained with a sampling density of 8 and a blocking number of 8 × 9. The Gabor features are extracted using a wavelength of 4 at four orientations ($\theta = \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$). **ALOI**[6]**:** It contains 110250 object images which can be divided into 1000 categories. We select the 10 types of objects from them, with a total of 1080 samples. **Caltech-20**[7]**:** It consists of 2386 object pictures. These images have 6 view features and can be divided into 20 categories. **Scene-15**[8]**:** It contains 4485 scene pictures with 15 categories.

Compared methods: We compare MRTMC with sixteen related MVC methods to verify the clustering performance of MRTMC, including Co-Regularized Spectral Clustering **(CoReg)** [56], Affinity Aggregation for Spectral Clustering **(AASC)** [15], Robust Multi-view Spectral Clustering **(RMSC)** [18], Adaptively Weighted Procrustes MVC **(AWP)** [57], Weighted Multi-view Spectral Clustering based on Spectral Perturbation **(WMSC)** [58], Multi-view Consensus Graph Clustering **(MCGC)** [20], Graph-based MVC **(GMC)** [21], Low-rank Tensor Constrained Multi-view Subspace Clustering **(LTMSC)** [32], Latent Multi-view Subspace Clustering **(LMSC)** [27], MVC via Tensor Multi-rank Minimization **(t-SVD-MSC)** [33], Hyper-Laplacian Regularized MVC (**HLR-M^{2}VS**) [34], Weighted Tensor-Nuclear Norm Minimization **(WTNNM)** [40], Generalized Nonconvex Low-rank Tensor Approximation **(GNLTA)** [45], MVSC via Adaptive Graph Learning **(AGLLFA)** [31], Robust Weighted Low-rank Tensor Approximation **(RWLTA)** [41], and Multi-view and Multi-order Graph Clustering **(MoMvGC)** [22].

Initialization and parameters setting: For all datasets, we apply the robust principal component analysis (PCA) [59] technique to divide the raw multi-view data into the latent clean multi-view data $X_c^{(v)}$ and noise data $X_n^{(v)}$. Then we use K-means based on singular value decomposition (KSVD) [60] for $X_n^{(v)}$ to obtain a reasonable initial value of the structural noise dictionary $D_s^{(v)} \in \mathbb{R}^{d_v \times k_s}$ and the coefficient $Z_s^{(v)} \in \mathbb{R}^{k_s \times n}$, where $k_s = \alpha \times c$, c is the number of clusters. And we set the initial value of the Gaussian noise as $E^{(v)} = 0$. For parameter settings, the proposed method contains regularization parameters λ_1 and λ_2, the penalty parameter ρ, the atom number of dictionary for each class α, and the nonconvex tensor nuclear norm parameters η and θ. In our experiments, we set the initial values of λ_1, λ_2, and ρ as 10^{-3}, which increase at a rate of 1.5 after each iteration. The maximum value of ρ is set to 10^{12}, and the maximum value of λ_1 and λ_2 are selected from the set of {1, 5, 10, 15, 20, 30, 40, 50}. Specifically, the values of parameters λ_1 and λ_2 in all test datasets are shown in Table 3. η, θ and α are empirically set to 4, 0.1, and 5, respectively. Moreover, a detailed analysis of these parameters can be found in Section 4.4. For the compared methods, all parameters are carefully tuned as suggested in the reference papers to achieve the highest ACC values.

Evaluation metrics: Clustering accuracy (ACC), normalized mutual information (NMI), and purity (PUR) are used to measure the clustering effect. The range of the three indicators is from 0 to 1, with higher values indicating better performance.

4.2. Clustering performance on simulated data

In the subsection, MSRC-V1, ORL, Yale and COIL-20 are used to test the clustering performance of the proposed method. The mixed noise is simulated by the following rules:

Case1 (**Gaussian noise**): The Gaussian noise with zero-mean and 0.1 variance is added.

Case2 (**Gaussian noise + salt-and-pepper noise**): On the basis of Case1, the salt and pepper noise with 0.1 intensity is added.

Case3 (**Gaussian noise + salt-and-pepper noise + block noise**): On the basis of Case2, several 5 × 5 or 8 × 8 black blocks are added to a view of 50% samples.

Case4 (**Gaussian noise + salt-and-pepper noise with different intensities**): All samples are corrupted by a combination of Gaussian noise and salt-and-pepper noise, with Gaussian noise variance and salt-and-pepper noise intensity randomly sampled from a uniform distribution $U(0.20, 0.30)$.

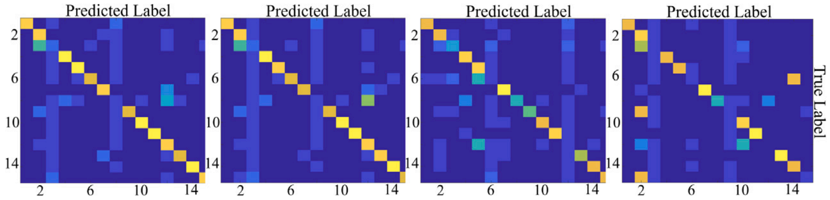
Tables 4–5 show the clustering results of MRTMC and the compared methods on four simulated datasets. The best result is **bold**, and the second-best is underlined. To see the clustering performance more intuitively, we visualize the confusion matrix as shown in Fig. 4 for Case3 of Yale, and use the standardized t-SNE [61] to visualize the

---

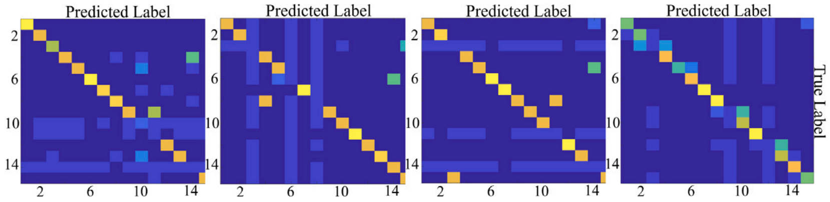
[3] http://www.uk.research.att.com/facedatabase.html.
[4] http://cvc.yale.edu/projects/yalefaces/yalefaces.html.
[5] http://www.cs.columbia.edu/CAVE/software/softlib/.
[6] https://aloi.science.uva.nl/.
[7] http://www.vision.caltech.edu/ImageDatasets/Caltech101/.
[8] http://www-cvr.ai.uiuc.edu/ponce_grp/data/.

Table 4
The clustering performance of different methods on simulated MSRC-V1 and ORL.

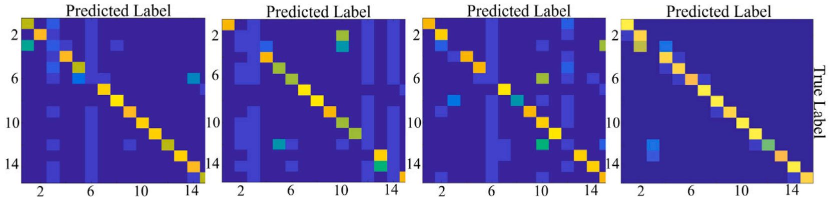
Datasets	Method	Case1			Case2			Case3			Case4		
		ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR
MSRC-V1	CoReg	0.8524	0.7440	0.8524	0.8143	0.6669	0.8143	0.7476	0.6200	0.7476	0.5810	0.3930	0.5810
	AASC	0.8381	0.7175	0.8381	0.8238	0.6969	0.8238	0.7524	0.6360	0.7571	0.4762	0.3010	0.5048
	RMSC	0.7571	0.6382	0.7571	0.7000	0.6274	0.7524	0.6619	0.5501	0.6667	0.5190	0.3479	0.5190
	AWP	0.8000	0.6897	0.8000	0.7905	0.6627	0.8333	0.6571	0.5741	0.7714	0.5905	0.3727	0.5905
	WMSC	0.8143	0.6933	0.8143	0.7429	0.6534	0.7476	0.7000	0.6207	0.7286	0.6381	0.4727	0.6381
	MCGC	0.7143	0.6057	0.7429	0.6905	0.5754	0.7048	0.6762	0.5354	0.6810	0.4381	0.2969	0.4476
	GMC	0.7857	0.6982	0.8286	0.7762	0.6787	0.8333	0.7476	0.6751	0.8571	0.2952	0.1958	0.3095
	LTMSC	0.8571	0.7301	0.8571	0.7619	0.6231	0.7619	0.7429	0.5922	0.7429	0.4667	0.3168	0.4810
	LMSC	0.6286	0.5253	0.6667	0.6095	0.4753	0.6095	0.5905	0.4429	0.7429	0.3476	0.1778	0.3524
	t-SVD-MSC	0.9762	0.9493	0.9762	0.9048	0.8248	0.9048	0.8571	0.7376	0.8571	0.8905	0.7935	0.8905
	HLR-M^{2}VS	0.9952	0.9892	0.9952	0.9667	0.9330	0.9667	0.9667	0.9330	0.9667	0.9286	0.8666	0.9286
	WTNNM	0.9667	0.9429	0.9667	0.9667	0.9349	0.9667	0.9619	0.9284	0.9619	0.9318	0.8819	0.9381
	GNLTA	0.9905	0.9784	0.9905	0.9810	0.9567	0.9810	0.9857	0.9676	0.9857	0.9571	0.9129	0.9571
	AGLLFA	0.8524	0.7385	0.8524	0.8429	0.7387	0.8429	0.8381	0.7052	0.6792	0.4524	0.2917	0.4524
	RWLTA	0.9571	0.9095	0.9571	0.9329	0.8724	0.9329	0.9276	0.8574	0.9276	0.9284	0.8854	0.9284
	MoMvGC	0.8000	0.6776	0.9000	0.7333	0.6318	0.7667	0.7190	0.6077	0.7571	0.4143	0.2462	0.4286
	MRTMC	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**0.9810**	**0.9602**	**0.9810**
ORL	CoReg	0.7450	0.8417	0.7600	0.7150	0.8356	0.7325	0.6375	0.8107	0.6850	0.5925	0.7455	0.6325
	AASC	0.6925	0.7831	0.7350	0.6575	0.7771	0.7175	0.6500	0.7804	0.7575	0.5500	0.6896	0.5900
	RMSC	0.5650	0.7153	0.5825	0.4575	0.6432	0.4850	0.4400	0.6255	0.4875	0.4325	0.6058	0.4925
	AWP	0.7325	0.8432	0.7675	0.6975	0.8177	0.7200	0.6625	0.8102	0.7175	0.5800	0.7420	0.6250
	WMSC	0.7425	0.8448	0.7650	0.7300	0.8447	0.7525	0.7225	0.8351	0.7450	0.6325	0.7873	0.6775
	MCGC	0.5850	0.7506	0.6825	0.5825	0.7155	0.7050	0.5475	0.6966	0.6375	0.4825	0.6353	0.5200
	GMC	0.5650	0.7208	0.8200	0.5450	0.6797	0.7850	0.5275	0.6794	0.7925	0.4950	0.6262	0.5775
	LTMSC	0.7775	0.8786	0.8225	0.7275	0.8554	0.7725	0.6950	0.8310	0.7500	0.7200	0.8470	0.7425
	LMSC	0.7350	0.8652	0.7825	0.7300	0.8427	0.7550	0.7100	0.8356	0.7425	0.5825	0.7438	0.6125
	t-SVD-MSC	0.9375	0.9728	0.9625	0.9000	0.9463	0.9300	0.8725	0.9422	0.9175	0.8525	0.9090	0.8625
	HLR-M^{2}VS	**0.9700**	**0.9906**	0.9750	0.9600	**0.9883**	0.9725	0.9425	0.9749	0.9525	0.9250	0.9616	0.9425
	WTNNM	0.9600	0.9833	0.9725	0.9350	0.9812	0.9500	0.9325	0.9669	0.9475	0.8875	0.9480	0.9050
	GNLTA	0.9550	0.9829	0.9675	0.9275	0.9779	0.9475	0.9250	0.9741	0.9450	0.9025	0.9521	0.9150
	AGLLFA	0.7450	0.8474	0.7725	0.6800	0.7902	0.6950	0.6525	0.7851	0.6875	0.6325	0.7747	0.6625
	RWLTA	0.9523	0.9817	0.9643	0.9425	0.9810	0.9590	0.9317	0.9713	0.9468	0.9270	0.9685	0.9430
	MoMvGC	0.6250	0.7827	0.6950	0.6100	0.7783	0.6850	0.5825	0.7497	0.6625	0.5575	0.7253	0.6375
	MRTMC	**0.9700**	0.9848	**0.9875**	**0.9625**	0.9828	**0.9825**	**0.9525**	**0.9792**	**0.9775**	**0.9625**	**0.9872**	**0.9750**



(a) RMSC (b) WMSC (c) MCGC (d) GMC



(e) LTMSC (f) t-SVD-MSC (g) HLR-M^{2}VS (h) WTNNM



(i) GNLTA (j) RWLTA (k) MoMvGC (l) MRTMC

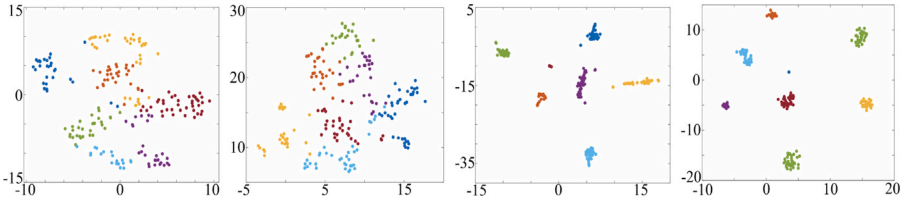
Fig. 4. Visualization of confusion matrices by different methods on Yale in Case3.

embedded results as shown in Fig. 5 for Case3 of MSRC-V1. Through the above operations, we can obtain the following conclusions:

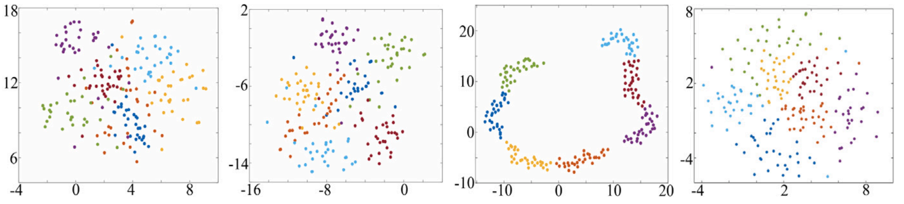
(1) In most cases, MRTMC outperforms baselines in terms of ACC, NMI, and PUR. Specifically, the ACC of the proposed model significant improve by 5.45%, 6.67%, 6.66%, and 5.45% compared to the sub-optimal method GNLTA in Case1, Case2, Case3, and Case4 on Yale dataset, respectively. Intuitively, the proposed method has clearer diagonal blocks than other compared methods from Fig. 4(a)–(l). It can

Table 5
The clustering performance of different methods on simulated Yale and COIL-20.

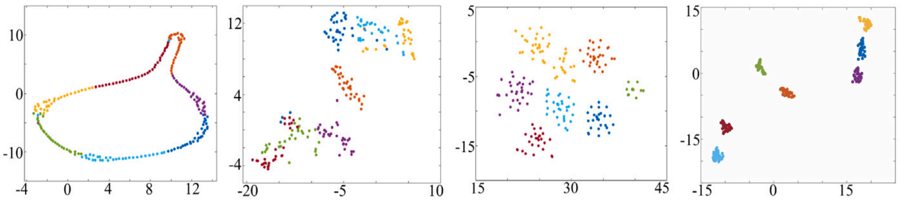
Datasets	Method	Case1			Case2			Case3			Case4		
		ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR
Yale	CoReg	0.6545	0.6538	0.7152	0.6424	0.6549	0.7091	0.6242	0.6321	0.6788	0.6364	0.6587	0.6424
	AASC	0.6000	0.5914	0.6424	0.5939	0.5948	0.6485	0.5273	0.5708	0.6061	0.5091	0.5354	0.5333
	RMSC	0.6667	0.6603	0.7030	0.6606	0.6514	0.6909	0.6535	0.6449	0.6887	0.5879	0.6463	0.6000
	AWP	0.6545	0.6652	0.7091	0.6121	0.6372	0.6667	0.5818	0.6141	0.6606	0.6061	0.6489	0.6364
	WMSC	0.6545	0.6415	0.6909	0.6485	0.6504	0.7030	0.6121	0.6184	0.6545	0.5818	0.6426	0.6061
	MCGC	0.6303	0.6268	0.6848	0.6182	0.6274	0.6727	0.6061	0.6273	0.6485	0.4970	0.5202	0.5152
	GMC	0.6909	0.7105	0.7758	0.6000	0.6289	0.7091	0.5455	0.6115	0.5576	0.5697	0.5783	0.5939
	LTMSC	0.6545	0.6851	0.7394	0.6242	0.6649	0.7212	0.6121	0.6361	0.6909	0.5152	0.4885	0.5152
	LMSC	0.6485	0.6785	0.7333	0.6303	0.6651	0.7152	0.6182	0.6533	0.7030	0.5455	0.5700	0.5576
	t-SVD-MSC	0.7212	0.7313	0.7697	0.6303	0.6673	0.7212	0.5939	0.6065	0.6182	0.6364	0.6781	0.6364
	HLR-M^{2}VS	0.7818	0.7733	0.7818	0.7273	0.7631	0.7273	0.6545	0.7197	0.6545	0.6182	0.6433	0.7212
	WTNNM	0.7242	0.7544	0.7242	0.7091	0.7380	0.7091	0.6061	0.6397	0.6061	0.6788	0.7221	0.6848
	GNLTA	0.7879	0.7988	0.7879	0.7515	0.7728	0.7515	0.7273	0.7602	0.7273	0.6970	0.6923	0.7030
	AGLLFA	0.7273	0.7360	0.7333	0.6848	0.6849	0.6970	0.6485	0.6946	0.6667	0.6545	0.6874	0.6606
	RWLTA	0.6891	0.7234	0.6915	0.6564	0.6824	0.6630	0.6339	0.6605	0.6370	0.6413	0.6797	0.6431
	MoMvGC	0.6848	0.7043	0.6848	0.6424	0.6581	0.6545	0.6364	0.6310	0.6424	0.6242	0.6027	0.6303
	MRTMC	**0.8424**	**0.8647**	**0.8727**	**0.8182**	**0.8664**	**0.8424**	**0.7939**	**0.8439**	**0.8485**	**0.7515**	**0.7951**	**0.7697**
COIL-20	CoReg	0.7458	0.8097	0.7688	0.7278	0.7962	0.7562	0.6813	0.7658	0.7299	0.6743	0.7802	0.7125
	AASC	0.6140	0.7181	0.7049	0.5826	0.6985	0.7097	0.5611	0.6514	0.6868	0.5917	0.6865	0.6299
	RMSC	0.6681	0.7817	0.7611	0.6521	0.7739	0.7319	0.6444	0.7587	0.7229	0.5181	0.5776	0.5549
	AWP	0.7090	0.8063	0.7583	0.6597	0.7782	0.7604	0.6382	0.7306	0.6882	0.6431	0.7452	0.6785
	WMSC	0.6750	0.7672	0.7583	0.6472	0.7606	0.7292	0.6174	0.7285	0.7278	0.6250	0.7633	0.6854
	MCGC	0.6569	0.7611	0.8090	0.6340	0.7359	0.8042	0.5306	0.6514	0.7521	0.5653	0.6541	0.6062
	GMC	0.7701	0.8597	0.8583	0.7271	0.8229	0.8472	0.6306	0.7869	0.7831	0.6403	0.7779	0.6903
	LTMSC	0.6944	0.7596	0.7264	0.5986	0.6596	0.6604	0.5514	0.5800	0.5993	0.6097	0.7102	0.6333
	LMSC	0.7028	0.7706	0.7111	0.6271	0.7438	0.6875	0.6194	0.6829	0.6458	0.6229	0.7114	0.6479
	t-SVD-MSC	0.7528	0.8270	0.8014	0.7438	0.8103	0.7708	0.7250	0.7726	0.7424	0.6861	0.7394	0.7271
	HLR-M^{2}VS	0.7799	0.8390	0.8125	0.7472	0.8261	0.7479	0.7417	0.8059	0.7604	0.7153	0.8143	0.7701
	WTNNM	0.8215	0.8876	0.8423	0.7979	0.8688	0.8174	0.7798	0.8637	0.8014	0.7646	0.8338	0.7910
	GNLTA	0.8167	0.8934	0.8389	0.8021	0.8701	0.8208	0.7924	0.8708	0.8139	0.7417	0.8154	0.7444
	AGLLFA	0.7424	0.8157	0.7472	0.7250	0.7902	0.7375	0.7028	0.7506	0.7132	0.6562	0.7093	0.6708
	RWLTA	**0.8890**	**0.9147**	**0.9014**	**0.8761**	**0.9161**	**0.8969**	**0.8251**	0.8540	**0.8403**	0.7763	0.8262	0.7937
	MoMvGC	0.8181	0.8832	0.8521	0.6743	0.7981	0.7292	0.6340	0.7140	0.6465	0.6438	0.7900	0.6889
	MRTMC	0.8431	0.8980	0.8576	0.8222	0.8805	0.8368	0.8056	**0.8825**	0.8347	**0.8076**	**0.8587**	**0.8090**



(a) RMSC (b) WMSC (c) MCGC (d) GMC



(e) LTMSC (f) t-SVD-MSC (g) HLR-M^{2}VS (h) WTNNM



(i) GNLTA (j) RWLTA (k) MoMvGC (l) MRTMC

Fig. 5. Visualization of the embedding results by different methods on MSRC-V1 in Case3.

be concluded that MRTMC is more robust to different mixed noise than other methods.

(2) Compared with the graph learning-based MVC methods, such as WMSC, MCGC and GMC, MRTMC achieves the best clustering performance. Specifically, in Case4 of the COIL-20 dataset, MRTMC significantly improves ACC, NMI, and PUR by 16.73%, 8.08%, and 11.87%,

Table 6
The clustering performance (ACC, NMI, and PUR) of different methods on the real-world datasets.

Method	AR			FERET			ALOI			Caltech-20			Scene-15		
	ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR
CoReg	0.2854	0.5576	0.3446	0.3921	0.7408	0.4393	0.9278	0.8708	0.9278	0.5101	0.5573	0.5784	0.4341	0.4206	0.4689
AASC	0.3062	0.5623	0.3700	0.3229	0.6585	0.4021	0.9250	0.8647	0.9250	0.5163	0.6157	0.6844	0.4484	0.4432	0.4923
RMSC	0.2154	0.5381	0.3008	0.3186	0.6851	0.3686	0.7667	0.7559	0.8407	0.5083	0.5553	0.7514	0.4305	0.4158	0.4698
AWP	0.2746	0.5468	0.3408	0.3786	0.7323	0.4243	0.8222	0.7895	0.9037	0.5520	0.5615	0.6316	0.3761	0.3550	0.4049
WMSC	0.2708	0.5610	0.3277	0.3693	0.7325	0.4129	0.8944	0.8257	0.8944	0.5025	0.5638	0.5838	0.4479	0.4521	0.4760
MCGC	0.2785	0.5379	0.3154	0.3714	0.7127	0.4171	0.9389	0.8789	0.9389	0.5754	0.4902	0.6686	0.3344	0.3137	0.3683
GMC	0.2162	0.4036	0.3069	0.2907	0.4875	0.6143	0.8537	0.7922	0.8537	0.4891	0.4234	0.6435	0.1953	0.1291	0.2089
LTMSC	0.6108	0.7906	0.6838	0.4329	0.7484	0.4700	0.9259	0.8642	0.9259	0.4991	0.5853	0.7791	0.3882	0.3673	0.2158
LMSC	0.4485	0.6781	0.4854	0.3764	0.7265	0.4200	0.8306	0.7877	0.8611	0.4594	0.5531	0.7672	0.3889	0.3598	0.4435
t-SVD-MSC	0.9031	0.9682	0.9278	0.6721	0.8665	0.7214	0.9231	0.8596	0.9231	0.4715	0.6472	0.5536	0.7501	0.7589	0.8147
HLR-M^{2}VS	0.6900	0.8261	0.6977	0.6257	0.8576	0.6486	0.9611	0.9279	0.9611	0.6039	0.6865	0.8479	0.7249	0.7176	0.7672
WTNNM	0.9054	0.9728	0.9231	0.7443	0.9036	0.7728	0.9250	0.8658	0.9250	0.5645	0.7143	0.8835	0.7507	0.7472	0.8087
GNLTA	0.8769	0.9585	0.9015	0.7407	0.9210	0.7779	**0.9852**	**0.9701**	**0.9852**	0.5524	0.7207	0.8722	0.8504	0.8609	0.8778
AGLLFA	0.5077	0.7117	0.5231	0.3607	0.7005	0.3800	0.9352	0.8722	0.9352	0.4531	0.4536	0.6953	0.3824	0.3696	0.4100
RWLTA	0.8578	0.9519	0.8841	0.7314	0.9091	0.7683	0.9564	0.9306	0.9564	0.6009	0.7188	0.8639	0.9337	**0.9137**	**0.9424**
MoMvGC	0.2785	0.4997	0.3092	0.3486	0.6181	0.3936	0.9454	0.9030	0.9454	0.6077	0.6336	0.7460	0.4087	0.3776	0.4441
MRTMC	**0.9292**	**0.9740**	**0.9608**	**0.7629**	**0.9245**	**0.8321**	0.9769	0.9546	0.9769	**0.6320**	**0.7810**	**0.9208**	**0.9344**	0.8837	0.9344

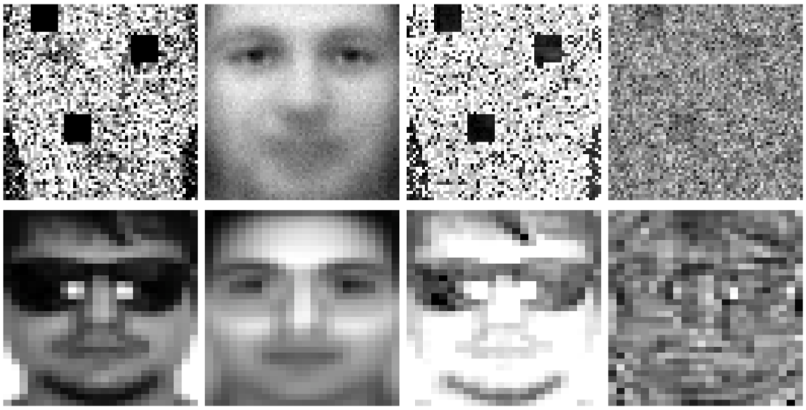

(a) Observed data (b) Recovered data (c) Structural noise (d) Gaussian noise

Fig. 6. Visualization of the recovered images, structural noise, and Gaussian noise on ORL with Case3 (first row) and the real-world dataset AR (second row).

respectively, compared to GMC. The graph learning-based methods directly construct similarity graphs from the raw multi-view data, but a single norm constraint cannot effectively depict the mixed noise, resulting in unreliable similarity graphs. In contrast, MRTMC takes into account both mixed noise and high-order correlation among views. From Fig. 5, we observe that MRTMC accurately classifies the samples into seven categories in Case3 of MSRC-V1 dataset.

(3) Compared with the subspace learning-based methods, such as t-SVD-MSC, HLR-M^{2}VS and WTNNM, MRTMC demonstrates outstanding capability. And one can see that the tensor-based methods achieve better clustering performance than the graph-based methods (such as WMSC, MCGC, and GMC). This may be because there is a strong similarity among multi-view data, and high-order correlation can better explore the consistency of data and obtain better clustering performance. As a comparison, since the proposed method can not only capture the global correlation among views but also explore the specific information within each view. And the performance of the proposed method MRTMC is better than the graph-based and tensor-based comparison methods. In addition, the proposed method has clearer cluster than other subspace-based MVC methods from Fig. 5(e)–(j). This is attributed to the fact that the mixed noise within each view can be adaptively learned and suppressed, thereby achieving the superior clustering performance.

4.3. Clustering performance on real-world data

In the subsection, AR, FERET, ALOI, Caltech-20, and Scene-15 are used to test the clustering performance of MRTMC. The above datasets contain complex structural noise, such as glasses, mask, and background as shown in Fig. 3(c) and (d). In Table 6, we list the clustering results of all methods. The best result is **bold**, and the second-best is underlined.

From Table 6, MRTMC demonstrates superior performance on the real datasets. Specially, in term of ACC, the proposed method shows significant improvements of 2.38%, 1.86%, and 2.43% over sub-optimal method on AR, FERET, and Caltech-20, respectively. Besides, the proposed method achieve promising result on ALOI and Scene-15. It is worth noting that graph learning-based MVC methods perform poorly on AR datasets. The may be attributed to the severe occlusion and varying illumination in AR dataset, which lead to unreliable similarity graphs constructed from the raw data. The reason why the proposed model is superior to other MVSC methods may be that mixed noise learning of the proposed method has better noise description ability than a single norm. Although the ALOI dataset is affected by environmental interference such as light conditions and rotation angles, all methods have strong clustering performance. This could be due to certain multi-view features that alleviate the impact of environmental interference.

Additionally, we have visualized the mixed noise separation as shown in Fig. 6. From the first line, the proposed model regards the black blocks and salt-and-pepper noise as structural noise and restores the outline of the face clearly. From the second line, the proposed model regards glasses and hair as structural noise and restores the occluded areas clearly. In summary, the features of the potential restored image are well preserved, and the structural noise can be separated

Table 7
The results of ablation experiments for the proposed method MRTMC.

Method	AR			FERET			ALOI			Caltech-20			Scene-15		
	ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR
MRTMC-(w/o)L_1	0.8385	0.9341	0.8985	0.3486	0.7065	0.3614	0.8435	0.7591	0.8435	0.4132	0.5666	0.7590	0.9144	0.8778	0.9144
MRTMC-(w/o)L_2	0.9062	0.9653	0.9446	0.7464	0.9157	0.8093	0.9667	0.9395	0.9667	0.5830	0.7462	0.9116	0.8685	0.8787	0.9010
MRTMC-(w/o)L_3	0.8909	0.9637	0.9377	0.7386	0.9145	0.8057	0.9639	0.9342	0.9639	0.5650	0.7392	0.8998	0.9248	0.8723	0.9248
MRTMC	**0.9292**	**0.9740**	**0.9608**	**0.7629**	**0.9245**	**0.8321**	**0.9769**	**0.9546**	**0.9769**	**0.6320**	**0.7810**	**0.9208**	**0.9344**	**0.8837**	**0.9344**

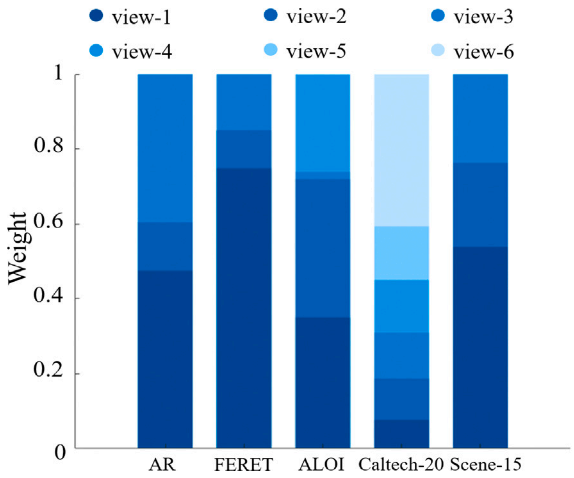


Fig. 7. The learned weights of different views for different datasets.

approximately. The proposed approach of representing the mixed noise brings valuable potential features, which is beneficial to clustering performance.

4.4. Discussions

Ablation study: To study the effectiveness of the different terms in proposed model, we perform experiments on three variants. For simplicity, L_1, L_2, and L_3 are used to represent adaptive dictionary learning for structural noise, nonconvex low-rank tensor approximation, and adaptive weighting strategy, respectively. Specifically, the model is degraded to MRTMC-(w/o)L_1 by removing the adaptive dictionary learning for structural noise. By replacing the nonconvex TNN with the traditional TNN, the model is degraded to MRTMC-(w/o)L_2. Giving each view the same weight instead of adaptive weighting learning, the model is degraded to MRTMC-(w/o)L_3. From the Table 7, MRTMC shows a significant improvement compared to MRTMC-(w/o)L_1. It indicates that the adaptive dictionary learning for structural noise is a crucial component of our method, leading to a significant enhancement of clustering performance.

Compared with the same weights (MRTMC-(w/o)L_3), the proposed method with the adaptive weights can achieve the higher clustering results, which shows the effectiveness of adaptive weight strategy for improving clustering performance. Fig. 7 presents the learned weights of our weighting strategy, which can adaptively determine the weights of different views for different datasets. For example, the weights of each view of Scene-15 data are 0.5393 (view-1), 0.2241 (view-2), and 0.2366 (view-3). In summary, the performance of the proposed model is better than all variants, which implies that each regularization term can enhance the performance of the model.

Effectiveness on adaptive dictionary learning: For the structural noise, we design the dictionary learning to adaptively adjust to varying structural noise characteristics across different datasets, which lies in two-fold:

- During the initialization phase, for different datasets, we use the robust PCA [59] to divide the raw multi-view data into the latent multi-view data and noise data. Then according to the characteristics of noise data, we use KSVD [60] to adaptively represent the noise data as the structural noise dictionary and the coefficient, resulting in the reasonable initialization.
- During the optimization phase, the proposed adaptive dictionary learning strategy can dynamically update the dictionary and representation coefficient of the structural noise of different datasets. Fig. 8 shows the process of adaptively learning the structure noise for different datasets. One can see that the structures of the learned dictionary and representation coefficient become clearer as the number of iterations increases, and the structural noise can be separated from the observed data to improve the robustness of clustering.

To quantitatively test the performance of dictionary learning, we compare the difference between the learned structural noise $\hat{X}_s^{(v)} = D_s{}^{(v)} Z_s{}^{(v)}$ and the real structural noise $X_s^{(v)}$ by

$$Err = \sum_{v=1}^{V} \frac{\|X_s^{(v)} - \hat{X}_s^{(v)}\|_F}{\|X_s^{(v)}\|_F}. \tag{32}$$

A smaller Err value indicates that the proposed adaptive dictionary learning strategy obtains the more accurate structural noise estimation. In Fig. 9, we plot the Err curve of structural noise on ORL dataset with Case3. We can see that the Err curve drops rapidly after a certain number of iterations and tends to be stable, which demonstrates that the adaptive dictionary learning can approximately learn the structured noise.

The effect of the block noise with different sizes: To test the effect of the block noise with different sizes on the clustering performance, taking the ORL dataset as an example, we test the following cases:

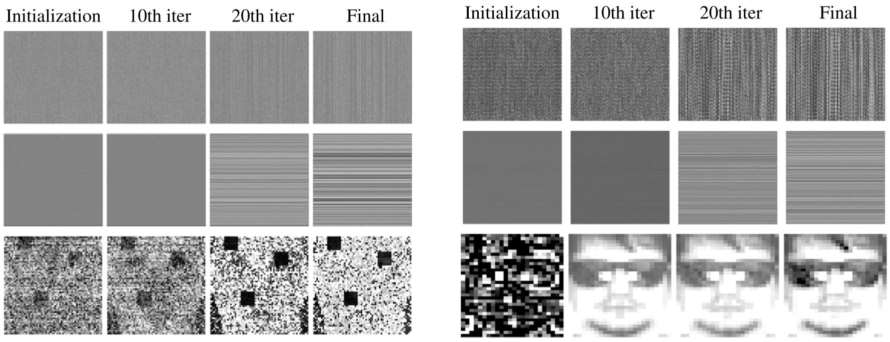
CaseA: On the basis of Case2, three 8 × 8 black blocks are added to a view of 50% samples.

CaseB: On the basis of Case2, two 20 × 20 black blocks are added to a view of 50% samples.

CaseC: On the basis of Case2, one 30 × 30 black block is added to a view of 50% samples.

Fig. 10 presents some samples contaminated by blocks of different sizes. Clearly, the larger the block noise, the more serious the data pollution. One can see that when the block noise is small, the restored latent image can well preserve the details and texture information of the image. As the block noise increases, although the quality of the restored image decreases, the main features of the image are preserved. Moreover, Table 8 lists the clustering performance of the proposed method and baselines in terms of different block noise cases. We observe that the proposed method achieves the best clustering performance and is robust to different block noise cases.

The impact of the regularization parameters λ_1 and λ_2: In the proposed method, regularization parameters λ_1 and λ_2 are used to balance the contributions of all terms in Eq. (6). Fixing the $k_s = 5c$, Fig. 11 provides ACC values of the proposed model with varying λ_1 and λ_2 on different datasets. The maximum value of λ_1 and λ_2 are selected from the set of {1, 5, 10, 15, 20, 30, 40, 50}. As the maximum value of λ_1 changes, the model performance changes significantly. We can conclude that λ_1 used to balance the contribution of low-rank term is sensitive to MRTMC. The change of the maximum value of λ_2 has little effect on the model performance, so it can be concluded that λ_2 which



(a) ORL with Case3

(b) real-world dataset AR

Fig. 8. Visualization of structural noise dictionary $D_s^{(v)}$ (first row), coefficient matrix $Z_s^{(v)}$ (second row), and structural noise (third row) on ORL with Case3 and real-world dataset AR as the number of iterations increases.

Table 8

The clustering performance of different methods in different block noise cases on ORL dataset.

Method	CaseA			CaseB			CaseC		
	ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR
CoReg	0.6375	0.8107	0.6850	0.6325	0.7721	0.6725	0.6150	0.7599	0.6525
AASC	0.6500	0.7804	0.7575	0.5950	0.7252	0.6425	0.5050	0.6572	0.5350
RMSC	0.4400	0.6255	0.4875	0.4325	0.6130	0.4925	0.3875	0.5538	0.4600
AWP	0.6625	0.8102	0.7175	0.6075	0.7295	0.6350	0.5875	0.7337	0.6225
WMSC	0.7225	0.8351	0.7450	0.6475	0.7707	0.6700	0.6250	0.7682	0.6625
MCGC	0.5475	0.6966	0.6375	0.5150	0.6728	0.5700	0.5050	0.6683	0.5825
GMC	0.5275	0.6794	0.7925	0.4875	0.6363	0.5825	0.4450	0.6083	0.5525
LTMSC	0.6950	0.8310	0.7500	0.6525	0.8013	0.7050	0.5850	0.7469	0.6100
LMSC	0.7100	0.8356	0.7425	0.6325	0.7601	0.6525	0.5525	0.7133	0.5750
t-SVD-MSC	0.8725	0.9422	0.9175	0.8750	0.9374	0.9025	0.8100	0.8992	0.8350
HLR-M^{2}VS	0.9425	0.9749	0.9525	0.9025	0.9530	0.9425	0.8725	0.9348	0.9025
WTNNM	0.9325	0.9669	0.9475	0.8925	0.9627	0.9175	0.8475	0.9217	0.8575
GNLTA	0.9250	0.9741	0.9450	0.8825	0.9364	0.8875	0.8575	0.9477	0.8925
AGLLFA	0.6525	0.7851	0.6875	0.6300	0.7731	0.6550	0.6075	0.7545	0.6350
RWLTA	0.9317	0.9713	0.9468	0.8850	0.9517	0.9075	0.8523	0.9329	0.8745
MoMvGC	0.5825	0.7497	0.6625	0.5925	0.7435	0.6425	0.5625	0.7370	0.6425
MRTMC	**0.9525**	**0.9792**	**0.9775**	**0.9275**	**0.9712**	**0.9475**	**0.8950**	**0.9435**	**0.9075**

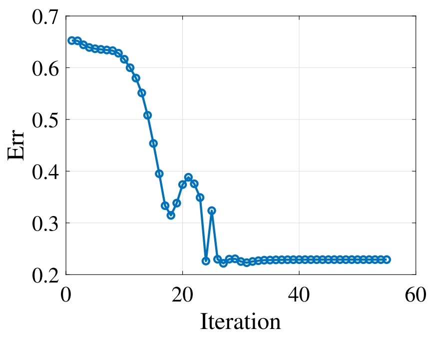


Fig. 9. The Err curve of structural noise on dataset ORL with Case3.

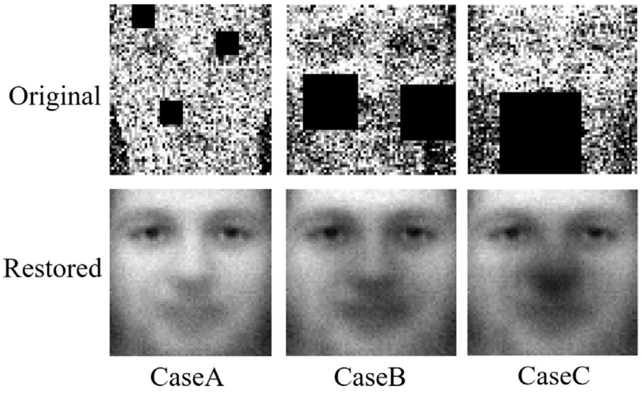


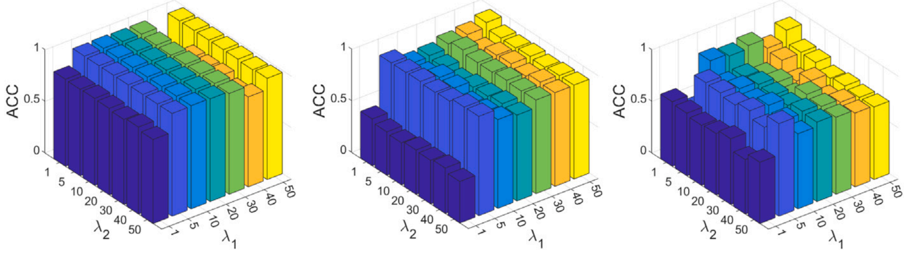
Fig. 10. The different block noise cases and the corresponding restored data on ORL dataset.

is used to balance the contribution of the structural noise sparsity is robust to MRTMC.

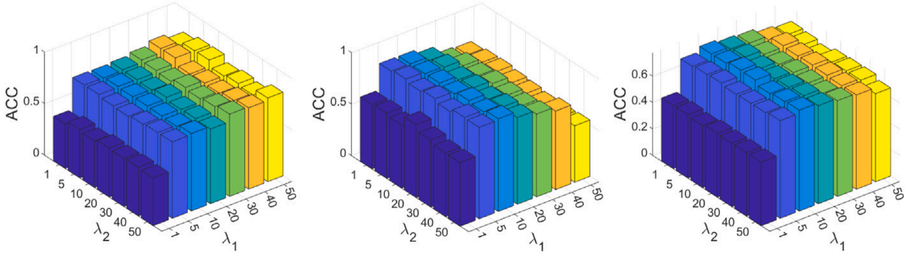
The impact of the atom number of dictionary α: The parameter α is used to balance the dictionary atoms and is selected from the set of {1, 5, 10, 15, 20}. Fig. 12 provides indices of the proposed model with varying α on different datasets. We can see that relative stable clustering performance can be achieved when α is within the set of {5, 10, 15, 20}. Considering the computational complexity, α is set 5 in all experiments.

The impact of the parameters η **and** θ **in the nonconvex tensor nuclear norm:** The nonconvex tensor nuclear norm includes parameters η and θ. To test their impact on the clustering results, taking datasets FERET and Scene-15 as examples, Fig. 13 shows the impact of parameters η and θ on clustering performance (i.e., ACC, NMI, and PUR values), respectively. From Fig. 13, we can see that the proposed method achieves the promising metrics in a relatively wide range of two parameters, which indicates that our methods are stable while changing η and θ. Therefore, we set $\eta = 4$ and $\theta = 0.1$ in all experiments.

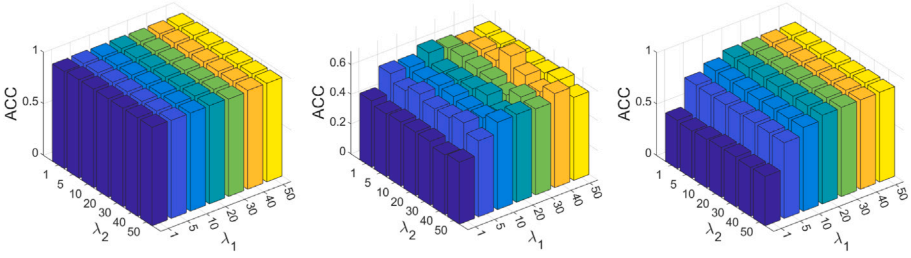
Comparison of running time: To further demonstrate the computational efficiency of the proposed method, Table 9 reports the running time of the proposed method and the compared methods. From Tables 4–6 and Table 9, we observe that the proposed method MRTMC can achieve the highest ACC, NMI, and PUR values within an acceptable running time in most cases. This shows that the proposed method is competitive in terms of computational efficiency.



(a) MSRC-Case3 (b) ORL-Case1 (c) Yale-Case1

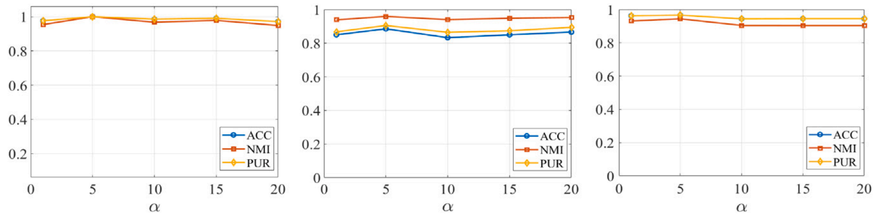


(d) COIL-Case1 (e) AR (f) FERET

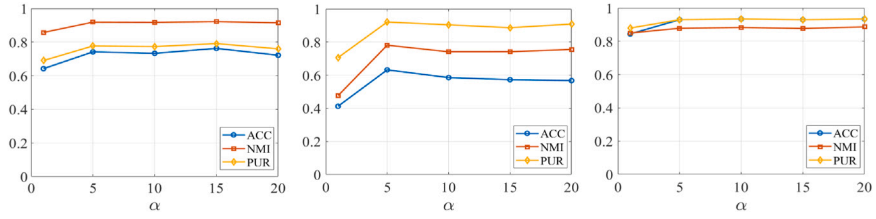


(g) ALOI (h) Caltech-20 (i) Scene-15

Fig. 11. ACC values of the proposed method with different parameters λ_1 (axis on the right) and λ_2 (axis on the left) on different datasets.



(a) MSRC-V1-Case1 (b) AR (c) ALOI

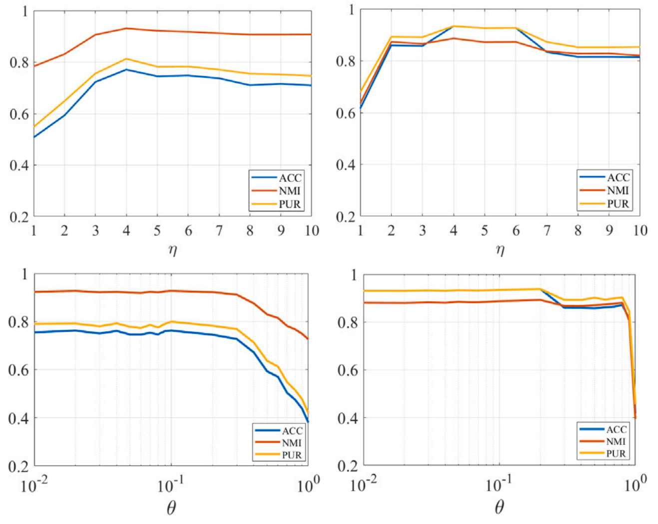


(d) FERET (e) Caltech-20 (f) Scene-15

Fig. 12. The clustering performance of the proposed model with varying α on (a) MSRC-V1, (b) AR, (c) AIOL, (d) FERET, (e) Caltech-20, and (f) Scene-15 datasets.

Table 9
Comparison of running times (seconds) on all the datasets.

Method	MSRC-V1	ORL	Yale	COIL-20	AR	FERET	ALOI	Caltech-20	Scene-15	Average
LTMSC	3.9438	31.2144	7.7366	157.2228	125.3813	165.9597	66.4640	670.8783	1833.6899	340.2768
LMSC	2.4209	15.4454	6.7208	66.5998	45.6185	56.1340	26.6348	238.6877	1509.7940	218.6729
t-SVD-MSC	1.1904	10.8259	4.4488	29.9339	16.1389	28.6948	10.0434	117.5262	162.3185	42.3468
HLR-M^{2}VS	1.4434	6.8569	2.3158	45.9249	41.8489	49.9118	39.9036	369.4876	1261.6286	202.1468
WTNNM	2.2146	16.7459	5.2575	50.2492	48.0626	39.2379	17.0176	233.0602	985.4711	155.2574
GNLTA	0.3214	0.4425	0.1199	2.7652	4.6005	3.8881	3.4018	27.5237	48.1953	10.1398
AGLLFA	0.4292	6.1409	1.6208	5.0536	22.3245	51.1857	2.4112	23.6487	50.7653	18.1756
RWLTA	0.7638	1.3053	0.3365	9.0607	14.3119	16.8936	13.4076	97.6755	117.9121	30.1852
MRTMC	1.0521	10.6151	4.2267	23.2636	26.0139	53.7062	19.3692	180.7157	313.9995	70.3291



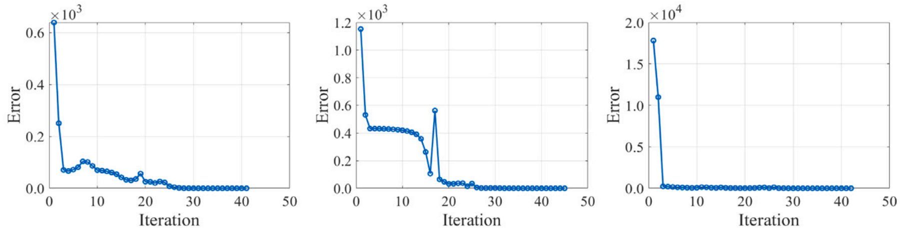
(a) FERET (b) Scene-15

Fig. 13. Clustering performance (ACC, NMI, and PUR values) with respect to the parameters η and θ.

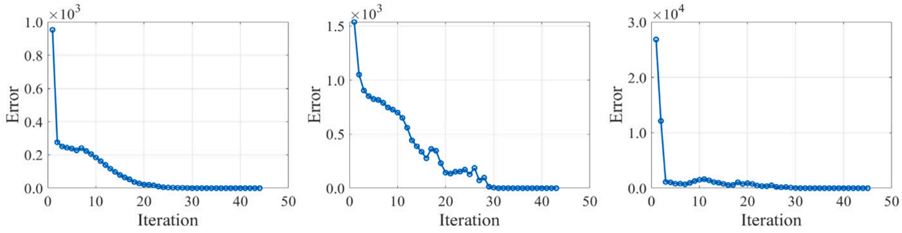
Numerical convergence: Theorem 1 demonstrates the theoretical convergence of the proposed MRTMC algorithm, and in this section we investigate the numerical convergence of the proposed MRTMC algorithm. The error curve ($\mathrm{Error} = \max\{\|\mathcal{Z}_s^{k+1} - \mathcal{Y}^{k+1}\|_\infty, \|\mathcal{D}_s^{k+1} - \mathcal{B}^{k+1}\|_\infty, \|\mathcal{Z}_c^{k+1} - \mathcal{G}^{k+1}\|_\infty\}$) is illustrated in Fig. 14. The abscissa represents the number of iterations, and the ordinate represents the maximum relative error. It can be observed that the relative error decreases with the increasing number of iterations, stabilizing at a small value after approximately 30 iterations. We conclude that the proposed algorithm is convergent numerically.

5. Conclusion

In this work, a new mixed-noise robust tensor multi-view clustering method via adaptive dictionary learning is proposed to improve the robustness of multi-view clustering. In MRTMC, the multi-view data is decomposed into three parts: latent clean multi-view data, structural noise, and Gaussian noise, where the structural noise with semantic information is modeled by adaptive dictionary learning. Moreover, the nonconvex tensor nuclear norm is introduced on the self-representation tensor to explore the high-order correlation among multiple views. In addition, an adaptive weighting strategy is used to distinguish the importance of different views. An efficient algorithm is developed within the ADMM framework to solve our model, and the theoretical convergence of the optimization algorithm is established. Experiments on simulated and real-world data illustrate the robustness and superiority of the proposed method over the state-of-the-art MVC methods.



(a) MSRC-V1-case1 (b) AR (c) ALOI



(d) FERET (e) Caltech-20 (f) Scene-15

Fig. 14. Relative error versus iterations of the proposed method on different datasets.

CRediT authorship contribution statement

Jing-Hua Yang: Writing – review & editing, Writing – original draft, Software, Methodology, Data curation. **Yi Zhou:** Visualization, Software, Methodology, Investigation. **Lefei Zhang:** Writing – review & editing, Supervision. **Heng-Chao Li:** Writing – review & editing, Supervision, Resources, Methodology.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (No. 12401605 and 62271418), and in part by the Natural Science Foundation of Sichuan Province (No. 2024NSFSC1467), Postdoctoral Fellowship Program of CPSF (No. GZC20232198), China Postdoctoral Science Foundation (No. 2024M752661), and Fundamental Research Funds for the Central Universities, China (No. 2682024CX017).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.inffus.2025.103322.

Data availability

Data will be made available on request.

References

[1] J. Zhao, X. Xie, X. Xu, S. Sun, Multi-view learning overview: Recent progress and new challenges, Inf. Fusion 38 (2017) 43–54.

[2] J. Xu, et al., Self-supervised discriminative feature learning for deep multi-view clustering, IEEE Trans. Knowl. Data Eng. 35 (7) (2023) 7470–7482.

[3] J. Chen, A. Huang, W. Gao, Y. Niu, T. Zhao, Joint shared-and-specific information for deep multi-view clustering, IEEE Trans. Circuits Syst. Video Technol. 33 (12) (2023) 7224–7235.

[4] X. Hu, X. Liu, W. Pedrycz, Q. Liao, Y. Shen, Y. Li, S. Wang, Multi-view fuzzy classification with subspace clustering and information granules, IEEE Trans. Knowl. Data Eng. 35 (11) (2023) 11642–11655.

[5] J. Zhou, F. Nie, X. Luo, X. He, Hierarchical bipartite graph based multi-view subspace clustering, Inf. Fusion 117 (2025) 102821.

[6] Q. Shen, Y. Chen, C. Zhang, Y. Tian, Y. Liang, Pick-and-place transform learning for fast multi-view clustering, IEEE Trans. Image Process. 33 (2024) 1272–1284.

[7] C. Zhang, D. Xu, C. Chen, M. Zhang, H. Li, Multi-relational multi-view clustering and its applications in cancer subtype identification, Inf. Fusion 117 (2025) 102831.

[8] C. Tang, et al., CGD: Multi-View clustering via cross-view graph diffusion, in: Proc. AAAI Conf. Artif. Intell., 2020, pp. 5924–5931.

[9] Y. Wang, D. Chang, Z. Fu, J. Wen, Y. Zhao, Incomplete multiview clustering via cross-view relation transfer, IEEE Trans. Circuits Syst. Video Technol. 33 (1) (2022) 367–378.

[10] G. Zhong, C.-M. Pun, Improved normalized cut for multi-view clustering, IEEE Trans. Pattern Anal. Mach. Intell. 44 (12) (2021) 10244–10251.

[11] N. Rappoport, R. Shamir, Multi-omic and multi-view clustering algorithms: review and cancer benchmark, Nucleic Acids Res. 46 (20) (2018) 10546–10562.

[12] Y. Xie, B. Lin, Y. Qu, C. Li, W. Zhang, L. Ma, Y. Wen, D. Tao, Joint deep multi-view learning for image clustering, IEEE Trans. Knowl. Data Eng. 33 (11) (2021) 3594–3606.

[13] X. Cao, C. Zhang, C. Zhou, H. Fu, H. Foroosh, Constrained multi-view video face clustering, IEEE Trans. Image Process. 24 (11) (2015) 4381–4393.

[14] M. Ding, X.-L. Zhao, J.-H. Yang, Z. Zhou, M.K. Ng, Bilateral tensor low-rank representation for insufficient observed samples in multidimensional image clustering and recovery, SIAM J. Imaging Sci. 18 (1) (2025) 20–59.

[15] H.-C. Huang, Y.-Y. Chuang, C.-S. Chen, Affinity aggregation for spectral clustering, in: Proc. IEEE Conf. Comput. Vision Pattern Recognit., 2012, pp. 773–780.

[16] H. Wang, G. Jiang, J. Peng, R. Deng, X. Fu, Towards adaptive consensus graph: multi-view clustering via graph collaboration, IEEE Trans. Multimed. 25 (2022) 6629–6641.

[17] K. Zhan, C. Niu, C. Chen, F. Nie, C. Zhang, Y. Yang, Graph structure fusion for multiview clustering, IEEE Trans. Knowl. Data Eng. 31 (10) (2019) 1984–1993.

[18] R. Xia, Y. Pan, L. Du, J. Yin, Robust multi-view spectral clustering via low-rank and sparse decomposition, in: Proc. AAAI Conf. Artif. Intell., 2014, pp. 2149–2155.

[19] F. Nie, J. Li, X. Li, Self-weighted multiview clustering with multiple graphs, in: Proc. Int. Joint Conf. Artif. Intell., 2017, pp. 2564–2570.

[20] K. Zhan, F. Nie, J. Wang, Y. Yang, Multiview consensus graph clustering, IEEE Trans. Image Process. 28 (3) (2019) 1261–1270.

[21] H. Wang, Y. Yang, B. Liu, GMC: Graph-based multi-view clustering, IEEE Trans. Knowl. Data Eng. 32 (6) (2020) 1116–1129.

[22] H. Xin, Z. Hao, Z. Sun, R. Wang, Z. Miao, F. Nie, Multi-view and multi-order graph clustering via constrained $l_{1,2}$-norm, Inf. Fusion 111 (2024) 102483.

[23] Y. Liang, D. Huang, C.-D. Wang, P.S. Yu, Multi-view graph learning by joint modeling of consistency and inconsistency, IEEE Trans. Neural Netw. Learn. Syst. 35 (2) (2024) 2848–2862.

[24] Y. Tan, Y. Liu, H. Wu, J. Lv, S. Huang, Metric multi-view graph clustering, in: Proc. AAAI Conf. Artif. Intell., 2023, pp. 9962–9970.

[25] B. Cheng, G. Liu, J. Wang, Z. Huang, S. Yan, Multi-task low-rank affinity pursuit for image segmentation, in: Proc. IEEE Int. Conf. Comput. Vis., 2011, pp. 2439–2446.

[26] H. Gao, F. Nie, X. Li, H. Huang, Multi-view subspace clustering, in: Proc. IEEE Int. Conf. Comput. Vis., 2015, pp. 4238–4246.

[27] C. Zhang, Q. Hu, H. Fu, P. Zhu, X. Cao, Latent multi-view subspace clustering, in: Proc. IEEE Conf. Comput. Vision Pattern Recognit., 2017, pp. 4333–4341.

[28] X. Cao, C. Zhang, H. Fu, S. Liu, H. Zhang, Diversity-induced multi-view subspace clustering, in: Proc. IEEE Conf. Comput. Vision Pattern Recognit., 2015, pp. 586–594.

[29] X. Wang, X. Guo, Z. Lei, C. Zhang, S.Z. Li, Exclusivity-consistency regularized multi-view subspace clustering, in: Proc. IEEE Conf. Comput. Vision Pattern Recognit., 2017, pp. 1–9.

[30] X. Cai, D. Huang, G.-Y. Zhang, C.-D. Wang, Seeking commonness and inconsistencies: A jointly smoothed approach to multi-view subspace clustering, Inf. Fusion 91 (2023) 364–375.

[31] C. Tang, K. Sun, C. Tang, X. Zheng, X. Liu, J.-J. Huang, W. Zhang, Multi-view subspace clustering via adaptive graph learning and late fusion alignment, Neural Netw. 165 (2023) 333–343.

[32] C. Zhang, H. Fu, S. Liu, G. Liu, X. Cao, Low-rank tensor constrained multiview subspace clustering, in: Proc. IEEE Int. Conf. Comput. Vis., 2015, pp. 1582–1590.

[33] Y. Xie, D. Tao, W. Zhang, Y. Liu, L. Zhang, Y. Qu, On unifying multi-view self-representations for clustering by tensor multi-rank minimization, Int. J. Comput. Vis. 126 (2018) 1157–1179.

[34] Y. Xie, W. Zhang, Y. Qu, L. Dai, D. Tao, Hyper-Laplacian regularized multilinear multiview self-representations for clustering and semisupervised learning, IEEE Trans. Cybern. 50 (2) (2020) 572–586.

[35] Y. Chen, X. Xiao, Y. Zhou, Jointly learning kernel representation tensor and affinity matrix for multi-view clustering, IEEE Trans. Multimed. 22 (8) (2020) 1985–1997.

[36] Y. Chen, X. Xiao, Z. Hua, Y. Zhou, Adaptive transition probability matrix learning for multiview spectral clustering, IEEE Trans. Neural Netw. Learn. Syst. 33 (9) (2022) 4712–4726.

[37] S. Wang, Y. Chen, Z. Lin, Y. Cen, Q. Cao, Robustness meets low-rankness: Unified entropy and tensor learning for multi-view subspace clustering, IEEE Trans. Circuits Syst. Video Technol. 33 (11) (2023) 6302–6316.

[38] M. Yin, J. Gao, S. Xie, Y. Guo, Multiview subspace clustering via tensorial t-product representation, IEEE Trans. Neural Netw. Learn. Syst. 30 (3) (2019) 851–864.

[39] Y. Chen, X. Xiao, C. Peng, G. Lu, Y. Zhou, Low-rank tensor graph learning for multi-view subspace clustering, IEEE Trans. Circuits Syst. Video Technol. 32 (1) (2021) 92–104.

[40] Q. Gao, W. Xia, Z. Wan, D. Xie, P. Zhang, Tensor-SVD based graph learning for multi-view subspace clustering, in: Proc. AAAI Conf. Artif. Intell., 2020, pp. 3930–3937.

[41] X. Pu, H. Che, B. Pan, M.-F. Leung, S. Wen, Robust weighted low-rank tensor approximation for multiview clustering with mixed noise, IEEE Trans. Comput. Soc. Syst. 11 (3) (2024) 3268–3285.

[42] Y. Liu, X. Zhang, G. Tang, D. Wang, Multi-view subspace clustering based on tensor schatten-p Norm, in: Proc. IEEE Int. Conf. Big Data, 2019, pp. 5048–5055.

[43] X. Sun, R. Zhu, M. Yang, X. Zhang, Y. Tang, Sliced sparse gradient induced multi-view subspace clustering via tensorial arctangent rank minimization, IEEE Trans. Knowl. Data Eng. 35 (7) (2023) 7483–7496.

[44] B. Pan, C. Li, H. Che, Nonconvex low-rank tensor approximation with graph and consistent regularizations for multi-view subspace learning, Neural Netw. 161 (2023) 638–658.

[45] Y. Chen, S. Wang, C. Peng, Z. Hua, Y. Zhou, Generalized nonconvex low-rank tensor approximation for multi-view subspace clustering, IEEE Trans. Image Process. 30 (2021) 4022–4035.

[46] W. Guo, H. Che, M.-F. Leung, Tensor-based adaptive consensus graph learning for multi-view clustering, IEEE Trans. Consum. Electr. 70 (2) (2024) 4767–4784.
[47] M.E. Kilmer, K. Braman, N. Hao, R.C. Hoover, 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.
[48] M.E. Kilmer, C.D. Martin, Factorization strategies for third-order tensors, Linear Algebra Appl. 435 (3) (2011) 641–658.
[49] M. Ding, J.H. Yang, X.L. Zhao, J. Zhuang, M.K. Ng, Nonconvex low-rank tensor representation for multi-view subspace clustering with insufficient observed samples, IEEE Trans. Knowl. Data Eng. (2025) 1–14.
[50] X.L. Zhao, J.H. Yang, T.H. Ma, T.X. Jiang, M.K. Ng, Tensor completion via complementary global, local, and nonlocal priors, IEEE Trans. Image Process. 31 (2022) 984–999.
[51] J.H. Yang, C. Chen, H.N. Dai, M. Ding, Z.B. Wu, Z.B. Zheng, Robust corrupted data recovery and clustering via generalized transformed tensor low-rank representation, IEEE Trans. Neural Netw. Learn. Syst. 35 (7) (2024) 8839–8853.
[52] C. Lu, C. Zhu, C. Xu, S. Yan, Z. Lin, Generalized singular value thresholding, in: Proc. AAAI Conf. Artif. Intell., 2015, pp. 1–7.
[53] C. Lu, J. Tang, S. Yan, Z. Lin, Nonconvex nonsmooth low rank minimization via iteratively reweighted nuclear norm, IEEE Trans. Image Process. 25 (2) (2016) 829–839.
[54] A. Martinez, R. Benavente, The AR face database, Rapp. Tech. 24 (1998).
[55] P. Phillips, H. Wechsler, J. Huang, P.J. Rauss, The FERET database and evaluation procedure for face-recognition algorithms, Image Vis. Comput. 16 (5) (1998) 295–306.
[56] A. Kumar, P. Rai, H. Daumé, Co-regularized multi-view spectral clustering, in: Proc. Int. Conf. Neural Inf. Process. Syst., 2011, pp. 1413–1421.
[57] F. Nie, L. Tian, X. Li, Multiview clustering via adaptively weighted procrustes, in: Proc. ACM SIGKDD Conf. Knowl. Discov. Data Mining, 2018, pp. 2022–2030.
[58] L. Zong, X. Zhang, X. Liu, H. Yu, Weighted multi-view spectral clustering based on spectral perturbation, in: Proc. AAAI Conf. Artif. Intell., 2018, pp. 1–8.
[59] E.J. Candès, X.D. Li, Y. Ma, J. Wright, Robust principal component analysis? J. ACM 58 (3) (2011) 1–37.
[60] M. Aharon, M. Elad, A. Bruckstein, K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation, IEEE Trans. Signal Process. 54 (11) (2006) 4311–4322.
[61] L. van der Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (86) (2008) 2579–2605.