Robust Corrupted Data Recovery and Clustering via Generalized Transformed Tensor Low-Rank Representation

Jing-Hua Yang, *Graduate Student Member, IEEE*, Chuan Chen, Hong-Ning Dai, *Senior Member, IEEE*, Meng Ding, Zhe-Bin Wu, and Zibin Zheng, *Senior Member, IEEE*

***Abstract*— Tensor analysis has received widespread attention in high-dimensional data learning. Unfortunately, the tensor data are often accompanied by arbitrary signal corruptions, including missing entries and sparse noise. How to recover the characteristics of the corrupted tensor data and make it compatible with the downstream clustering task remains a challenging problem. In this article, we study a generalized transformed tensor low-rank representation (TTLRR) model for simultaneously recovering and clustering the corrupted tensor data. The core idea is to find the latent low-rank tensor structure from the corrupted measurements using the transformed tensor singular value decomposition (SVD). Theoretically, we prove that TTLRR can recover the clean tensor data with a high probability guarantee under mild conditions. Furthermore, by using the transform adaptively learning from the data itself, the proposed TTLRR model can approximately represent and exploit the intrinsic subspace and seek out the cluster structure of the tensor data precisely. An effective algorithm is designed to solve the proposed model under the alternating direction method of multipliers (ADMMs) algorithm framework. The effectiveness and superiority of the proposed method against the compared methods are showcased over different tasks, including video/face data recovery and face/object/scene data clustering.**

***Index Terms*— Recoverability guarantee, tensor data recovery, tensor subspace clustering, transformed tensor low-rank representation (TTLRR).**

Manuscript received 2 January 2022; revised 1 August 2022; accepted 14 October 2022. This work was supported in part by the Key-Area Research and Development Program of Guangdong Province under Grant 2020B010165003; in part by the National Natural Science Foundation of China under Grant 62176269 and Grant 12201522; in part by the Macao Science and Technology Development Fund, Macao Funding Scheme for Key Research and Development Projects, under Grant 0025/2019/AKP; and in part by the Innovative Research Foundation of Ship General Performance under Grant 25622112. *(Corresponding authors: Chuan Chen; Hong-Ning Dai.)*

Jing-Hua Yang is with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510275, China, and also with the School of Computer Science and Engineering, Faculty of Innovation Engineering, Macau University of Science and Technology, Taipa 999078, Macau (e-mail: yangjinghua110@126.com).

Chuan Chen and Zhe-Bin Wu are with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510275, China (e-mail: chenchuan@mail.sysu.edu.cn; wuzhb6@mail2.sysu.edu.cn).

Hong-Ning Dai is with the Department of Computer Science, Hong Kong Baptist University, Hong Kong (e-mail: hndai@ieee.org).

Meng Ding is with the School of Mathematics, Southwest Jiaotong University, Chengdu 610031, China (e-mail: dingmeng56@163.com).

Zibin Zheng is with the School of Software Engineering, Sun Yat-sen University, Zhuhai 519082, China (e-mail: zhzibin@mail.sysu.edu.cn).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TNNLS.2022.3215983.

Digital Object Identifier 10.1109/TNNLS.2022.3215983

I. Introduction

HIGH-DIMENSIONAL data recovery and data clustering [1], [2], [3], [4], [5], [6], [7] are two fundamental tasks in data analysis. To solve the two problems, low-rank representation methods that map the high-dimensional tensor into a low-dimensional latent subspace have achieved promising performance in various applications, such as data mining [8], [9], [10], [11], hybrid system identification [12], and computer vision [13], [14], [15], [16], [17]. Due to the existence of interference during the equipment acquisition and transmission, the observed data are usually incomplete and corrupted by the noise in real scenarios [18], [19], [20], [21], thereby highly degrading their recovery and clustering performance. Therefore, it is important to investigate the recovery and clustering problem of the tensor data from the sparsely noisy observations with missing values.

In the past few years, matrix low-rank representation (MLRR) methods have been widely applied in data recovery and clustering fields. MLRR methods consider estimating the original low-rank matrix component from its arbitrary sparse corruptions or incomplete observations. For example, Candès et al. [22] studied the robust principal component analysis approach for separating the low-rank and sparse parts from the contaminated matrix. Liu et al. [23] focused on recovering the low-rank matrix that lies in a union of multiple subspaces from the corrupted observation. In data clustering fields, Elhamifar and Vidal [24] utilized a sparse optimization program to cluster data points lying in the low-dimensional subspaces. Liu et al. [25] used the low-rank representation to identify the subspace structures and segment the samples into their respective subspaces. Meanwhile, many variants of MLRR have been further studied. Lu et al. [1] proposed a robust subspace segmentation (RSS) with the help of the least square regression. Tang et al. [26] studied the structure-constrained low-rank representation to improve MLRR for the disjoint subspace segmentation. Yan et al. [27] learned a linear projection matrix to transform the original data into a low-dimensional subspace for class separability by optimizing the between-class distance and the within-class variability. Though these studies focus on obtaining a low-rank matrix or building a good affinity matrix, they are required to flatten the data into a vector for recovery or clustering, in which the vectorization operator destroys the intrinsic structure of the original data. On the contrary, data in practical applications, such as video, color images, and traffic data [28], [29], are

usually distributed in high-dimensional space with inherent structural characteristics.

To preserve the intrinsic structure of data, the tensor-based low-rank representation (TLRR) approaches have gradually emerged. For instance, Zhang et al. [30] studied the low-rank regularized heterogeneous tensor decomposition model to obtain the lowest-rank representation of underlying data for subspace clustering. Johnson et al. [31] designed a two-way optimization model for the submodule clustering technique using the weighted tensor nuclear norm (TNN). Moreover, Chang et al. [32] adopted multiple orthogonal projections to find the low-dimensional representation of the original data, thereby resulting in the excellent classification performance. Zhou et al. [33] introduced the conventional TNN to constrain the low-rank tensor representation of data. Zhang et al. [34] approximated the low-rank component from the corrupted data by the convex envelop of Tucker rank. Chen et al. [35] used the nonconvex low-rank tensor approximation to characterize the low-rankness of underlying data in data clustering. Recently, deep learning has been studied for data clustering [36], [37], [38], [39], [40], [41], [42], [43]. To be specific, many works employ the common deep autoencoder [39], [40], [41] to learn the low-dimensional embedding representation and then apply the spectral clustering or k-means on the learned affinity matrix to obtain the clustering result. By integrating the representation learning and clustering results into a unified framework, convolutional neural network (CNN)-based clustering method [36], [43], information theory-based clustering methods [37], [44], contrastive learning-based clustering method [38], have been studied. Due to the powerful learning ability, the deep clustering methods can capture the complex linear and nonlinear structure embedded in the input data. However, the deep learning clustering methods cannot well handle the recovery and clustering of high-dimensional but small-scale data. In addition, they lack the theoretical guarantee for data recovery and clustering.

The above-mentioned methods usually hold the assumption that the observation data are complete, which leads to unsatisfactory results when dealing with incomplete data. To cope with the issue, the two-step strategy is brought up naturally. It works by using the low-rank tensor estimation methods first to restore data and then clustering the estimated underlying data. Among these approaches, abundant tensor recovery methods are proposed due to the diversiform definitions of tensor rank. For instance, Liu et al. [45] proposed a low-rank tensor completion method by using the convex surrogate of the Tucker rank [46], i.e., the sum of nuclear norms (SNN). Zhang and Aeron [47] studied the tensor completion problem by the TNN, which is the convex surrogate of the tensor tubal rank. To address higher dimensional data, tensor network decomposition methods have received attention. For example, Chen et al. [14] studied the robust tensor completion problem based on tensor train rank by dimensional augmentation. Owing to the advantage of tensor ring rank in dealing with high-dimensional data, Huang et al. [48] applied the tensor ring structure for incomplete high-dimensional data completion. Liu et al. [49] proposed the robust tensor completion method based on fully connected tensor network decomposition. To recover the tensor data corrupted by sparse errors, Lu et al. [50] applied TNN in the tensor robust principal component (TRPCA) model to recover the estimated low-rank data. For these two-step strategies, the characteristics learned from the separated low-rank tensor recovery might not be suitable for the downstream clustering tasks. To analyze the corrupted data effectively, Francis et al. [51] focused on the incomplete image data clustering by incorporating submodule clustering with the matrix completion algorithm. However, in real-world applications (e.g., the recommendation and system identification problem), due to the lower resolution of the camera and the damage to the sensor arrays, the observed tensor data is often accompanied by arbitrary signal corruptions, including missing entries and sparse noises [18], [19], [52], [53]. Therefore, the problem of simultaneous recovery and clustering for the incomplete and noisy data is more general while being a crucial task.

In this work, we propose a novel TTLRR method for recovering and clustering the observed tensor data with missing entries and sparse corruptions. In particular, we formulate the following model:

$$\begin{aligned} &\min_{\mathcal{Z},\mathcal{E}} \|\mathcal{Z}\|_{\mathrm{TTNN}} + \lambda\|\mathcal{E}\|_1 \\ &\text{s.t. } P_{\Omega}(\mathcal{X}) = P_{\Omega}(\mathcal{A} \diamond_{\Phi} \mathcal{Z} + \mathcal{E}) \end{aligned} \tag{1}$$

where $\lambda > 0$ is a parameter, $\mathcal{Z} \in \mathbb{R}^{n_2 \times n_2 \times n_3}$ is the underlying low-rank representation tensor, $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is the dictionary, i.e., the linear tensor representation, $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is the observed tensor with incomplete and noisy data, and $\mathcal{E} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ represents the sparse noise. Here, $\|\cdot\|_{\mathrm{TTNN}}$ denotes transformed TNN (TTNN), and $\|\cdot\|_1$ denotes the tensor l_1 norm (please see the notations in Section II). $\diamond_{\Phi}$ is the Φ-product of the tensor, see Definition 1. $P_{\Omega}(\cdot)$ is a projection operator, and Ω denotes the index of observed entries. For incomplete corrupted data, the proposed TTLRR method performs the tensor recovery and clustering simultaneously. Fig. 1 depicts the framework of the proposed TTLRR.

Theoretically, under mild conditions, we establish the recoverability guarantee of the proposed model. The proposed model can recover the clean data $\mathcal{A} \diamond_{\Phi} \mathcal{Z}$ under the guarantee of high probability. For data clustering, we show the tensor block-diagonal structure–property of the representation tensor when the data are completely restored. The tensor block-diagonal structure implies the underlying tensor subspaces, and thus TTLRR can achieve satisfactory clustering performance. To the best of our knowledge, it is the first approach to simultaneously recover and cluster both incomplete and noisy data with the theoretical guarantee.

According to the alternating direction method of multipliers (ADMMs) algorithmic framework, we design an effective algorithm for solving the proposed model. Extensive experiments on different datasets show the effectiveness and robustness of the proposed method for recovering and clustering the corrupted data under different incomplete and noise settings. In particular, the results obtained by using the adaptive unitary transform learned from given data are better than the predefined transform.

The main contributions of this work are summarized as follows.

1) *Generalized Tensor Model Framework:* We consider a unified tensor model framework for recovering and clustering the incomplete data with sparse noise, which is a more common and challenging task in many real-world applications. In the proposed framework, we arrange and

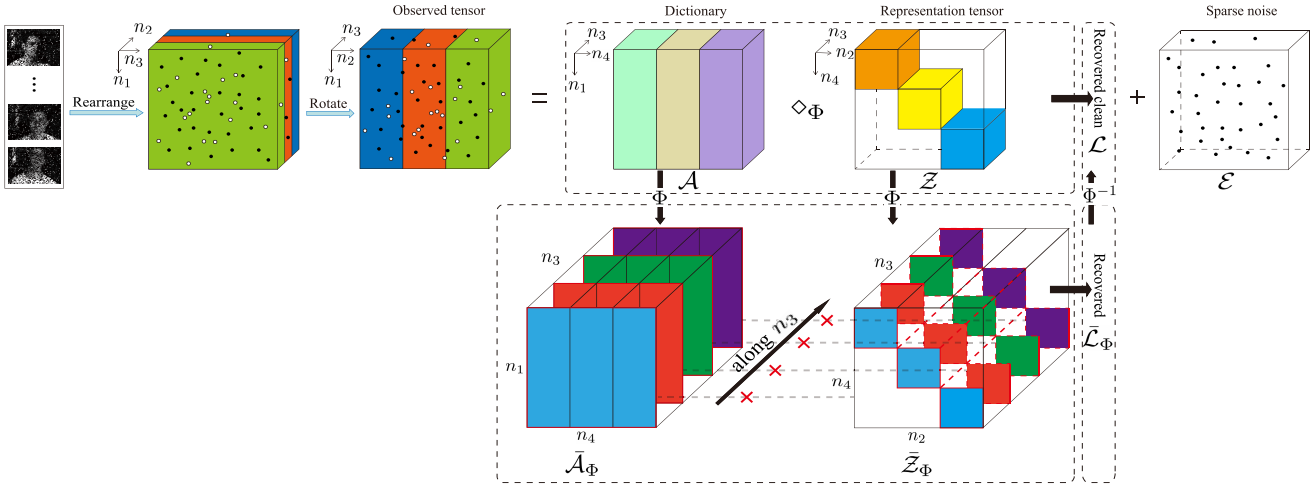


Fig. 1. Framework of the proposed method for high-dimensional data recovery and clustering. We first merge different data to construct the third-order tensor and then rotate it. Next, the transformed tensor low-rank representation (TTLRR)-based tensor optimization problem is utilized to obtain the optimal subspace representation. Finally, we can simultaneously obtain the recovered data by $\mathcal{A} \diamond_\Phi \mathcal{Z}$ and the clustering results by performing the spectral clustering tool on $\mathcal{Z}$. Specifically, to explore the inherent high-order correlations, dictionary $\mathcal{A}$ and representation tensor $\mathcal{Z}$ are converted into Φ space, i.e., the transformed tensor $\bar{\mathcal{A}}_\Phi$ and $\bar{\mathcal{Z}}_\Phi$, by using the transform Φ to each tube along the third dimension. In the transformed Φ space, $\bar{\mathcal{A}}_\Phi$, and $\bar{\mathcal{Z}}_\Phi$ perform the slice-by-slice standard matrix product to obtain tensor $\bar{\mathcal{L}}_\Phi$, and then $\mathcal{L}$ can be further obtained by performing the inverse transform Φ^H on $\bar{\mathcal{L}}_\Phi$.

rotate the observed data as a third-order tensor. Compared with the traditional matrix data representation, the tensor data representation can well preserve the intrinsic structure of the data.

2) *Transformed Tensor Low-Rank Representation:* We propose a TTLRR-based tensor recovery and subspace clustering method. We employ the unitary transform along the third dimension of the coefficient tensor $\mathcal{Z}$ and introduce the TTNN to characterize its low-rankness. Compared with the conventional discrete Fourier transform, using the suitable transform, e.g., the unitary transform learned by the given data, TTLRR can achieve a lower rank tensor representation and thus deeply explore the global correlations hidden in the data.
3) *Recoverability and Clusterability Guarantees:* We theoretically analyze the recoverability of the proposed model and explore the tensor block-diagonal structure–property of the optimal solution. Specifically, we establish the optimal solution of the TTLRR model under mild conditions. Meanwhile, the tensor block-diagonal structure of the obtained optimal solution further ensures the promising clustering performance. The experiments demonstrate the effectiveness of the proposed method.

The organization of this work is as follows. Section II presents some notations and preliminaries. Some related works are listed in Section III. Section IV introduces the proposed model and establishes the theoretical guarantee. Section V designs an effective algorithm for solving the optimization model. Sections VI and VII present the experimental results of two learning tasks, data recovery and data clustering, respectively. Section VIII summarizes the work.

II. Notations and Preliminaries

We introduce some notations and preliminaries used in this article.

We use the calligraphy letter $\mathcal{Z}$ to denote the tensor, the upper case letter $\boldsymbol{Z}$ to denote the matrix, the bold lower case letter $\boldsymbol{z}$ to denote the vector, and the lower case letter z to denote the scalar. For a third-order tensor $\mathcal{Z} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, its (i, j, k)th entry is denoted as $\mathcal{Z}_{i,j,k}$, and we use the MATLAB notation $\mathcal{Z}(i, :, :)$, $\mathcal{Z}(:, j, :)$, and $\mathcal{Z}(:, :, k)$ to denote the ith horizontal, jth lateral, and kth frontal slice, respectively. For convenience, we denote the frontal slice $\mathcal{Z}(:, :, k)$ by $\mathcal{Z}^{(k)}$, and the tube by $\mathcal{Z}(i, j, :)$. The inner product of two tensors in $\mathbb{R}^{n_1 \times n_2 \times n_3}$ is defined as $\langle \mathcal{Y}, \mathcal{Z} \rangle = \sum_{k=1}^{n_3} \langle \mathcal{Y}^{(k)}, \mathcal{Z}^{(k)} \rangle$.

Algorithm 1 Unitary Transform-Based Φ-Product [54]

Input: $\mathcal{A} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$, $\mathcal{B} \in \mathbb{C}^{n_2 \times l \times n_3}$, and $\Phi \in \mathbb{C}^{n_3 \times n_3}$.
Output: $\mathcal{C} = \mathcal{A} \diamond_\Phi \mathcal{B} \in \mathbb{C}^{n_1 \times l \times n_3}$.

1: Compute $\bar{\mathcal{A}}_\Phi = \Phi[\mathcal{A}]$ and $\bar{\mathcal{B}}_\Phi = \Phi[\mathcal{B}]$;
2: The each frontal slice of $\bar{\mathcal{C}}$ can be obtained by
3: $\bar{\mathcal{C}}_\Phi^{(k)} = \bar{\mathcal{A}}_\Phi^{(k)} \bar{\mathcal{B}}_\Phi^{(k)}$, $k = 1, \cdots, n_3$;
4: Obtain $\mathcal{C} = \Phi^{-1}[\bar{\mathcal{C}}_\Phi]$.

Now, we show some norms used in this work. For a third-order tensor, the l_1 norm is $\|\mathcal{Z}\|_1 = \sum_{i,j,k} |\mathcal{Z}_{i,j,k}|$, the Frobenius norm is $\|\mathcal{Z}\|_F = (\sum_{i,j,k} |\mathcal{Z}_{i,j,k}|^2)^{1/2}$, the infinity norm is defined as $\|\mathcal{Z}\|_\infty = \max_{i,j,k} |\mathcal{Z}_{i,j,k}|$. For a matrix, the nuclear norm is $\|\boldsymbol{Z}\|_* = \sum \sigma(\boldsymbol{Z})$, where $\sigma(\boldsymbol{Z})$ is the singular value of $\boldsymbol{Z}$.

A. Transformed Tensor Nuclear Norm

Let $\Phi \in \mathbb{C}^{n_3 \times n_3}$ be an unitary matrix satisfying $\Phi\Phi^H = \Phi^H\Phi = \boldsymbol{I}_{n_3}$, where Φ^H is the conjugate transpose of Φ and $\boldsymbol{I}_{n_3} \in \mathbb{R}^{n_3 \times n_3}$ is the identity matrix. We denote the unitary transform of $\mathcal{Z}$ by $\bar{\mathcal{Z}}_\Phi$, i.e., $\bar{\mathcal{Z}}_\Phi = \Phi(\mathcal{Z}, [], 3)$, along the third dimension. In particular, $\bar{\mathcal{Z}}_\Phi$ is the product of Φ and all tubes of $\mathcal{Z}$, that is,

$$\bar{\mathcal{Z}}_\Phi(i, j, :) = \Phi(\mathcal{Z}(i, j, :)). \tag{2}$$

$\mathcal{Z}$ can be obtained by the inverse unitary transform of $\bar{\mathcal{Z}}_\Phi$ along the third dimension, i.e., $\mathcal{Z} = \Phi^H(\bar{\mathcal{Z}}_\Phi, [], 3)$. For convenience, we denote $\Phi(\mathcal{Z}, [], 3)$ and $\Phi^H(\bar{\mathcal{Z}}_\Phi, [], 3)$ by $\Phi[\mathcal{Z}]$ and $\Phi^H[\bar{\mathcal{Z}}_\Phi]$, respectively.

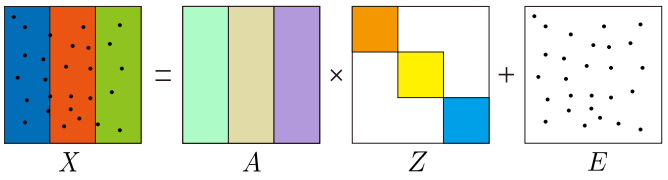


Fig. 2. Matrix-based low-rank representation.

The block diagonal matrix of $\bar{\mathcal{Z}}_\Phi$ has the following form:

$$\text{blockdiag}(\bar{\mathcal{Z}}_\Phi) = \begin{pmatrix} \bar{\mathcal{Z}}_\Phi^{(1)} & & \\ & \ddots & \\ & & \bar{\mathcal{Z}}_\Phi^{(n_3)} \end{pmatrix} \tag{3}$$

where $\bar{\mathcal{Z}}_\Phi^{(k)}$ denotes the kth frontal slice of $\bar{\mathcal{Z}}_\Phi$. Moreover, the tensor can be reduced by the fold operator

$$\text{fold}(\text{blockdiag}(\bar{\mathcal{Z}}_\Phi)) = \bar{\mathcal{Z}}_\Phi. \tag{4}$$

To better understand the definition of TTNN, we introduce some related definitions.

Definition 1 (Φ-Product): [54]: Let $\mathcal{A} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$ and $\mathcal{B} \in \mathbb{C}^{n_2 \times n_4 \times n_3}$, then the Φ-product of $\mathcal{A}$ and $\mathcal{B}$ is

$$\mathcal{C} = \mathcal{A} \diamond_\Phi \mathcal{B} = \Phi^H[\text{fold}(\text{blockdiag}(\bar{\mathcal{A}}_\Phi) \times \text{blockdiag}(\bar{\mathcal{B}}_\Phi))]$$

where $\mathcal{C} \in \mathbb{C}^{n_1 \times n_4 \times n_3}$ and $\times$ is the standard matrix product. Algorithm 1 presents the computation process of the unitary transform-based Φ-product.

Definition 2 (Conjugate Transpose): [54]: For tensor $\mathcal{Z} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$, the conjugate transpose $\mathcal{Z}^H$ can be obtained by $\mathcal{Z}^H = \Phi^H[\text{fold}(\text{blockdiag}(\bar{\mathcal{Z}}_\Phi)^H)]$.

Definition 3 (Identify Tensor): A tensor $\mathcal{I}_\Phi \in \mathbb{C}^{n \times n \times n_3}$ is called identify tensor if $\mathcal{I}_\Phi = \Phi^H[\mathcal{O}]$ and every frontal slice of $\mathcal{O} \in \mathbb{C}^{n \times n \times n_3}$ is the identity matrix with size $n \times n$.

Definition 4 (Orthogonal Tensor): [54]: Under transform Φ, if

$$\mathcal{U}^H \diamond_\Phi \mathcal{U} = \mathcal{U} \diamond_\Phi \mathcal{U}^H = \mathcal{I}_\Phi \tag{5}$$

then the tensor $\mathcal{U} \in \mathbb{C}^{n \times n \times n_3}$ is orthogonal.

Definition 5 (f-Diagonal Tensor): A tensor is called f-diagonal if each of its frontal slices is a diagonal matrix.

Definition 6 (T-TSVD): [54]: Let $\mathcal{Z} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$, then $\mathcal{Z}$ has the unitary transform-based tensor singular value decomposition (T-TSVD) as

$$\mathcal{Z} = \mathcal{U} \diamond_\Phi \mathcal{S} \diamond_\Phi \mathcal{V}^H \tag{6}$$

where $\mathcal{U} \in \mathbb{C}^{n_1 \times n_1 \times n_3}$ and $\mathcal{V} \in \mathbb{C}^{n_2 \times n_2 \times n_3}$ are orthogonal tensors, and $\mathcal{S} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$ is a diagonal tensor. The details of T-TSVD are presented in Algorithm 2.

Definition 7 (TTNN): [54]: Given $\mathcal{Z} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$, its TTNN is $\|\mathcal{Z}\|_{\text{TTNN}} = \sum_{k=1}^{n_3} \|\bar{\mathcal{Z}}_\Phi^{(k)}\|_*$, where $\|\bar{\mathcal{Z}}_\Phi^{(k)}\|_*$ is the nuclear norm of $\bar{\mathcal{Z}}_\Phi^{(k)}$.

Definition 8 (Transformed Tubal Rank): [54]: For tensor $\mathcal{Z} = \mathcal{U} \diamond_\Phi \mathcal{S} \diamond_\Phi \mathcal{V}^H$, the number of nonzero singular tubes of $\mathcal{S}$ is defined as the transformed tubal rank $\text{rank}_t(\mathcal{Z})$, that is,

$$\text{rank}_t(\mathcal{Z}) = \#\{i : \mathcal{S}(i, i, :) \neq 0\} \tag{7}$$

where # denotes the cardinality of a set.

Algorithm 2 T-TSVD Based on Φ-Product

Input: $\mathcal{Z} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$.
Output: $\mathcal{U}$, $\mathcal{S}$, and $\mathcal{V}$.
1: Compute $\bar{\mathcal{Z}}_\Phi = \Phi[\mathcal{Z}]$;
2: **for** $k = 1, 2, \ldots, n_3$ **do**
3: $[\boldsymbol{U}, \boldsymbol{S}, \boldsymbol{V}] = \text{SVD}(\bar{\mathcal{Z}}_\Phi^{(k)})$;
4: $\bar{\mathcal{U}}_\Phi^{(k)} = \boldsymbol{U}$, $\bar{\mathcal{S}}_\Phi^{(k)} = \boldsymbol{S}$, $\bar{\mathcal{V}}_\Phi^{(k)} = \boldsymbol{V}$;
5: **end for**
6: Compute $\mathcal{U} = \Phi^H[\bar{\mathcal{U}}_\Phi]$, $\mathcal{S} = \Phi^H[\bar{\mathcal{S}}_\Phi]$, and $\mathcal{V} = \Phi^H[\bar{\mathcal{V}}_\Phi]$.

III. Related Works

The low-rank representation is utilized as a powerful tool to capture the global structure of data, which has been extensively studied. It can be roughly divided into two categories: MLRR and TLRR.

A. Matrix-Based Low-Rank Representation

Given a 2-D dataset $\{\boldsymbol{X}_j\}_{j=1}^{n_2}$, where $\boldsymbol{X}_j \in \mathbb{R}^{n_1 \times n_3}$ and n_2 is the number of data samples. The MLRR method usually maps the 2-D data $\boldsymbol{X}_j \in \mathbb{R}^{n_1 \times n_3}$ into a vector $\boldsymbol{x}_j \in \mathbb{R}^{n_1 n_3}$, then forms a data matrix $\boldsymbol{X} \in \mathbb{R}^{n_1 n_3 \times n_2}$. Since the high-dimensional data can be well represented in a low-dimensional space, the columns of $\boldsymbol{X}$ are drawn from a union of K subspaces $\{K_k\}_{k=1}^K$, and K_k concludes m_k sample points with $\Sigma_{k=1}^K m_k = n_2$.

The general matrix-based low-rank representation model is

$$\begin{aligned} &\min_{\boldsymbol{Z}} \ R(\boldsymbol{Z}) + \lambda \|\boldsymbol{X} - \boldsymbol{A}\boldsymbol{Z}\|_1 \\ &\text{s.t. } \boldsymbol{X} = \boldsymbol{A}\boldsymbol{Z} + \boldsymbol{E} \end{aligned} \tag{8}$$

where $\boldsymbol{Z}$ is the coefficient matrix, $\boldsymbol{A}$ is the given dictionary, and $\boldsymbol{E}$ is the sparse noise (see Fig. 2 for illustration). The first term $R(\boldsymbol{Z})$ denotes the regularization of the coefficient matrix, and the second term utilizes the l_1-norm to improve the robustness for modeling the outlier $\boldsymbol{E}$. Existing works have proposed a variety of methods for computing the representation coefficient $\boldsymbol{Z}$ by imposing different $R(\boldsymbol{Z})$ regularizations. For example, the work in [24] implied a sparse constraint on the coefficient matrix by $\|\boldsymbol{Z}\|_1$. The work in [25] aimed to find a low-rank coefficient by nuclear norm $\|\boldsymbol{Z}\|_*$. Brbic and Kopriva [55] combined the advantages of both sparse and low-rank constraints for subspace clustering. For solving corrupted observations by sparse errors, Liu et al. [23] studied to recover a low-rank matrix that lies on a union of multiple subspaces. MLRR can cluster the data samples $\boldsymbol{X}$ by seeking the low-rank/sparse matrix $\boldsymbol{X}$ with respect to the given dictionary $\boldsymbol{A}$.

Although MLRR works in data recovery and clustering to a certain extent, it can only handle 2-D data and cannot work for high-dimensional tensor data. Besides, MLRR reshapes the 2-D data into a vector-valued matrix, and hence, the performance would be not satisfactory because of destroying the data structure and losing the spatial information. Therefore, one hopes to explore the high-order correlation of observed data from the tensor aspect.

B. Tensor-Based Low-Rank Representation

The tensor-based representation method is to arrange all data together into a tensor without destroying the intrinsic

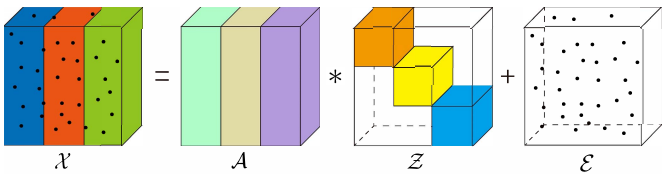


Fig. 3. TLRR.

structure of the original data. Specifically, we consider a third-order tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, where each sample $\mathcal{X}(:, j, :) \in \mathbb{R}^{n_1 \times n_3}$ is placed into the lateral slice of $\mathcal{X}$, and n_2 is the number of data samples.

Existing tensor-based methods employ the tensor decomposition (e.g., Tucker decomposition [30] and tensor singular value decomposition (SVD) [30], [33]) on the data $\mathcal{X}$ to find the subspace representation by the predefined or learned dictionary. Here, we briefly introduce the tensor SVD-based tensor representation method. Given a tensor dictionary $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, there exists a representation tensor $\mathcal{Z} \in \mathbb{R}^{n_2 \times n_2 \times n_3}$ that satisfies $\mathcal{X} = \mathcal{A} * \mathcal{Z} + \mathcal{E}$ (see Fig. 3), where $\mathcal{E}$ is the sparse tensor noise, and $*$ denotes the conventional Fourier transform-based tensor product [33]. Therefore, the general TLRR model is

$$\begin{aligned} &\min_{\mathcal{Z}} \ R(\mathcal{Z}) + \lambda \|\mathcal{X} - \mathcal{A} * \mathcal{Z}\|_1 \\ &\text{s.t. } \mathcal{X} = \mathcal{A} * \mathcal{Z} + \mathcal{E} \end{aligned} \tag{9}$$

where $R(\mathcal{Z})$ denotes the regularized term of $\mathcal{Z}$. Zhou et al. [33] utilized TNN to explore the low-tubal-rankness of the coefficient $\mathcal{Z}$ that can exactly recover the underlying tensor data and cluster them as well. Johnson et al. [31] introduced the weighted TNN to enhance the tensor sparsity. When the dictionary $\mathcal{A}$ is the identity tensor, model (9) reduces to the TRPCA problem [50]. Once a compact representation $\mathcal{Z}_\star$ is obtained, the similarity matrix of samples can be established by the frontal slice $\mathcal{Z}_\star^{(k)} \in \mathbb{R}^{n_2 \times n_2}$. Then, an affinity matrix can be computed by $\hat{\mathbf{Z}} = (1/2n_3)\Sigma_{k=1}^{n_3}(|\mathcal{Z}_\star^{(k)}| + |(\mathcal{Z}_\star^{(k)})^H|)$, and the spectral clustering approach (NCut, Ratio) is performed to obtain the clustering result.

Besides the sparse noise, the data usually contain missing entries. For incomplete data, based on different tensor ranks, many low-rank tensor completion methods have been proposed. Based on the Tucker model [46], Liu et al. [45] used its convex surrogate SNN to formulate the low-rank tensor completion model. Based on the tensor tubal rank [56], Zhang et al. [47] proposed a tensor completion method by minimizing its convex surrogate TNN. The works in [55] and [58] considered recovering the clean tensor under the case where both missing values and sparse noise exist.

Although the aforementioned matrix and tensor-based clustering methods have achieved great success in data analysis, there still exists some room for further improvement. For example, some works [54], [57] only focused on tensor data recovery while other work [33] studied the tensor recovery and clustering problem with only considering the sparse corruption. In the real scenario, data acquired from different sources come up with missing entries and sparse noise concurrently. These limitations motivate us to design a unified tensor low-rank representation framework of learning clean tensor data from the corrupted data and meanwhile doing the clustering task.

IV. Proposed TTLRR Model for Incomplete Data With Sparse Noise

We first present the model TTLRR for recovering and clustering the corrupted data under incomplete samples. Then, we present the theoretical analysis of the recoverability and clusterability guarantees.

A. Model Formulation

To explore the global correlation of samples and preserve the intrinsic structure of high-dimensional data, we follow the idea of TLRR, i.e., grouping a set of n_2 samples $\{\boldsymbol{X}_j \in \mathbb{R}^{n_1 \times n_3}\}_{j=1}^{n_2}$ into a tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$.

In this work, we study the transformed tensor low-rank representation for the incomplete data corrupted by sparse noise and propose the following TTLRR model:

$$\begin{aligned} &\min_{\mathcal{Z},\mathcal{E}} \ \|\mathcal{Z}\|_{\text{TTNN}} + \lambda \|\mathcal{E}\|_1 \\ &\text{s.t. } P_\Omega(\mathcal{X}) = P_\Omega(\mathcal{A} \diamond_\Phi \mathcal{Z} + \mathcal{E}). \end{aligned} \tag{10}$$

TTLRR consists of two terms. The first term $\|\mathcal{Z}\|_{\text{TTNN}}$ is TTNN of the representation tensor $\mathcal{Z}$ to characterize the potential low-rank structures of high-dimensional data. The second term $\|\mathcal{E}\|_1$ is used to constrain the sparse noise $\mathcal{E}$, aiming to reduce the effects of sparse noise on clustering.

Note that the optimization problems in works [33], [51], [55], and [58] can be seen as special cases of our TTLRR. For example, our model (10) is the transformed robust tensor completion [54] when the dictionary $\mathcal{A}$ remains an identity tensor; TTLRR also degenerates into the TLRR-based tensor clustering problem [33] when Ω is the whole indices set and Φ is the discrete Fourier transform. In summary, the advantages of TTLRR are as follows.

1) Compared with matrix-based subspace methods, TTLRR preserves the *intrinsic structure* of the original data and fully explores the *high-order correlation* of the tensor data.
2) Compared with existing tensor-based subspace methods, TTLRR considers a more *generalized and realistic* situation. Moreover, we use the TTNN obtained by the suitable unitary transform to characterize the low-rankness of the representation tensor and obtain a lower rank representation to improve the recovery and clustering performance.
3) The proposed low-rank tensor representation method can simultaneously achieve recovery and clustering with *theoretical guarantee*. Specifically, TTLRR recovers the underlying low-rank data via $\mathcal{A} \diamond_\Phi \mathcal{Z}$. Meanwhile, TTLRR can learn the similarity between samples via $\mathcal{Z}$ and cluster them by establishing the affinity matrix.

Remark: Note that the unitary transform is important for the performance of TTLRR. With a suitable transformation, a lower tubal-rank tensor can be learned by TTNN. Particularly, the unitary transform learned by given data, i.e., the data-dependent transform, is more flexible and superior than predefined transformations.

In Section IV-B, we will establish the theoretical performance guarantee in Section IV-B and demonstrate the effectiveness and robustness of TTLRR under various sampling rates and noise ratios in Sections VI and VII.

B. Theoretical Analysis

Here, we demonstrate that TTLRR (10) can exactly obtain the clean low-rank tensor $\mathcal{L} = \mathcal{A} \diamond_\Phi \mathcal{Z}_*$ with theoretical guarantee under mild conditions. Then, after recovering the clean tensor data, we theoretically prove that the minimizer $\mathcal{Z}_*$ has the block-diagonal structure. This suggests the satisfactory clustering performance.

In order to analyze how to exactly obtain the minimizer $(\mathcal{Z}_*, \mathcal{E}_*)$ of (10), we need to establish the transformed tensor incoherence conditions on $\mathcal{L} = \mathcal{A} \diamond_\Phi \mathcal{Z}$ to ensure $\mathcal{L}$ not be sparse, as discussed in [54]. We denote $\mathring{\mathfrak{e}}_i \in \mathbb{R}^{n_1 \times 1 \times n_3}$ as the tensor column basis with its $(i, 1, 1)$th entry equaling 1 and the rest equaling 0 and $\mathring{\mathfrak{e}}_j \in \mathbb{R}^{n_2 \times 1 \times n_3}$ as a tensor with its $(j, 1, 1)$th entry equaling to 1 and the rest equaling to 0. The tensor incoherence parameters μ_1 and μ_2 are defined as

$$\mu_1(\mathcal{L}) = \frac{n_2 n_3}{r} \max_{j=1,2,\ldots,n_2} \|\mathcal{V}^H \diamond_\Phi \mathring{\mathfrak{e}}_j\|_F^2$$
$$\mu_2(\mathcal{L}) = \frac{n_1 n_3}{r} \max_{i=1,2,\ldots,n_1} \|\mathcal{U}^H \diamond_\Phi \mathring{\mathfrak{e}}_i\|_F^2$$

where $r = \text{rank}_t(\mathcal{L})$, and $\mathcal{U} \diamond_\Phi \mathcal{S} \diamond_\Phi \mathcal{V}^H$ is the T-TSVD of $\mathcal{L}$. As TTLRR involves the dictionary $\mathcal{A}$, so we require the definition of a new incoherence condition for $\mathcal{A}$. Let $r_\mathcal{A} = \text{rank}_t(\mathcal{A})$. Then, the incoherence parameter related to $\mathcal{A}$ is defined as follows:

$$\mu_1^\mathcal{A}(\mathcal{L}) = \mu_1(\mathcal{L}) \max_{i=1,2,\ldots,n_1} \|\mathcal{U}_\mathcal{A}^H \diamond_\Phi \mathring{\mathfrak{e}}_i\|_F^2 = \frac{r_\mathcal{A}}{n_1 n_3} \mu_1(\mathcal{L}) \mu_2(\mathcal{A}).$$

Now, we present the convex model (10) that can exactly obtain the low-rank representation tensor $\mathcal{Z}_*$ and the sparse component $\mathcal{E}_*$ under mild conditions. Let $n_{(1)} = \max(n_1, n_2)$ and $n_{(2)} = \min(n_1, n_2)$. We also define a projection operator $\mathcal{P}_\mathcal{U}(\mathcal{B}) = \mathcal{U} \diamond_\Phi \mathcal{U}^H \diamond_\Phi \mathcal{B}$.

Theorem 1 (Main Result): Assume that $\mathcal{L} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ satisfies the incoherence conditions, the observation set $\mathbf{\Omega}$ is uniformly distributed in all cardinality sets $m = \rho n_1 n_2 n_3$, and each observed sample is independently damaged with probability γ. Suppose that $\mathcal{P}_{\mathcal{U}_\mathcal{A}}(\mathcal{U}) = \mathcal{U}$ and the ranks of $\bar{\mathcal{A}}^{(k)}$ $(k = 1, \ldots, n_3)$ are equal. Let $\mu^\mathcal{A} = \max\{\mu_2(\mathcal{L}), \mu_1^\mathcal{A}(\mathcal{L})\}$. Then, when $\lambda = 1/\sqrt{\rho n_{(1)} n_3}$, $(\mathcal{Z}_*, \mathcal{E}_*)$ is the unique optimal solution with probability at least $1 - 4(n_{(1)} n_3)^{-8}$, provided that

$$\text{rank}_t(\mathcal{L}) \leq \frac{c_r n_{(2)}}{\mu^\mathcal{A} \log(n_{(1)} n_3)} \quad \text{and} \quad \gamma \leq c_\gamma$$

where c_r and c_γ are positive constants.

The proof of Theorem 1 is presented in the Supplementary Material. Theorem 1 states that under certain conditions, TTLRR (10) can exactly obtain the underlying low-rank tensor $\mathcal{L} = \mathcal{A} \diamond_\Phi \mathcal{Z}_*$ with high probability.

After obtaining the clean tensor data by Theorem 1, we theoretically illustrate the clustering performance. For simplicity, we directly consider the tensor data to be complete and noise free. Then, the proposed model is generated as

$$\begin{aligned} &\min_\mathcal{Z} \ \|\mathcal{Z}\|_{\text{TTNN}} \\ &\text{s.t. } \mathcal{L} = \mathcal{A} \diamond_\Phi \mathcal{Z}. \end{aligned} \tag{11}$$

Now, we study the internal structure of the minimizer of the optimization problem (11). Here, we give the concept of tensor subspace.

Definition 9 (Tensor Subspace): [33]: The tensor subspace is defined as the set $\mathcal{K} = \{\mathcal{Y} | \mathcal{Y} = \mathcal{D} \diamond_\Phi \mathcal{C}, \forall \mathcal{C} \in \mathbb{R}^{k \times 1 \times n_3}\}$, where $\mathcal{D}_{(k)}$ is the kth lateral slice of $\mathcal{D} \in \mathbb{R}^{n_1 \times k \times n_3}$ and denotes the basis of $\mathcal{K}$. The set $\{\mathcal{D}_{(1)}, \ldots, \mathcal{D}_{(k)}\}$ is linearly independent, i.e., there is not a nonzero $\mathcal{C}$ satisfying $\mathcal{D} \diamond_\Phi \mathcal{C} = \mathbf{0}$.

The following theorem guarantees that the minimizer $\mathcal{Z}_*$ admits the tensor block-diagonal structure.

Theorem 2 (Tensor Block-Diagonal Property): For $\mathcal{L} = [\mathcal{L}_1, \ldots, \mathcal{L}_k]$, each sample $(\mathcal{L}_j(:, i, :))$ in $\mathcal{L}_j \in \mathbb{R}^{n_1 \times m_j \times n_3}$ comes from the jth tensor subspace $\mathcal{K}_j$ whose k_j basis components $\mathcal{A}_j \in \mathbb{R}^{n_1 \times k_j \times n_3}$. If the tensor space $\{\mathcal{K}_1, \ldots, \mathcal{K}_k\}$ are independent, then the minimizer $\mathcal{Z}_*$ of the problem (11) has the block-diagonal structure.

The proof of Theorem 2 is also presented in the Supplementary Material. Theorem 2 demonstrates the block-diagonal structure of $\mathcal{Z}_*$, which is crucial for clustering since this property directly demonstrates the tensor subspace in which a specific sample is located. Therefore, if the conditions in Theorem 2 are satisfied, the minimizer $\mathcal{Z}_*$ would be with block-diagonal structure, therefore, theoretically ensuring an accurate clustering.

V. Optimization Technique

We propose an effective algorithm for solving our model (10) and analyze its convergence.

A. Optimization Algorithm

As ADMM is deft at handling complex optimization problems by decomposing them into individual solvable subproblems, we adopt the ADMM to calculate the optimization model (10).

By introducing two auxiliary variables $\mathcal{Q} \in \mathbb{R}^{n_2 \times n_2 \times n_3}$ and $\mathcal{G} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, we covert (10) to the following problem:

$$\begin{aligned} &\min_{\mathcal{Z}, \mathcal{E}} \ \|\mathcal{Z}\|_{\text{TTNN}} + \lambda \|\mathcal{E}\|_1 \\ &\text{s.t. } \mathcal{Z} = \mathcal{Q}, \mathcal{M} = \mathcal{A} \diamond_\Phi \mathcal{Q} + \mathcal{G} \\ &\qquad \mathcal{E} = \mathcal{G}, P_\Omega(\mathcal{X}) = P_\Omega(\mathcal{M}). \end{aligned} \tag{12}$$

The corresponding augmented Lagrangian function of (12) is

$$\begin{aligned} &\mathcal{L}(\mathcal{Z}, \mathcal{E}, \mathcal{M}, \mathcal{Q}, \mathcal{G}) \\ &= \|\mathcal{Z}\|_{\text{TTNN}} + \lambda \|\mathcal{E}\|_1 + \Psi_\Omega(\mathcal{X} - \mathcal{M}) + \langle \mathcal{T}, \mathcal{Z} - \mathcal{Q} \rangle \\ &\quad + \frac{\beta_1}{2} \|\mathcal{Z} - \mathcal{Q}\|_F^2 + \langle \mathcal{H}, \mathcal{E} - \mathcal{G} \rangle + \frac{\beta_2}{2} \|\mathcal{E} - \mathcal{G}\|_F^2 \\ &\quad + \langle \mathcal{N}, \mathcal{M} - \mathcal{A} \diamond_\Phi \mathcal{Q} - \mathcal{G} \rangle + \frac{\beta_3}{2} \|\mathcal{M} - \mathcal{A} \diamond_\Phi \mathcal{Q} - \mathcal{G}\|_F^2 \end{aligned} \tag{13}$$

where $\mathcal{T}$, $\mathcal{H}$, and $\mathcal{N}$ denote Lagrangian multipliers, and β_1, β_2, and β_3 are penalty parameters. The $\Psi_\Omega(\cdot)$ denotes the indicator function

$$\Psi_\Omega(\mathcal{X} - \mathcal{M}) = \begin{cases} 0, & \text{if } \mathcal{X}_\Omega = \mathcal{M}_\Omega, \\ \infty, & \text{otherwise.} \end{cases}$$

Since all the variables can be separated into $(\mathcal{Z}, \mathcal{E}, \mathcal{M})$ and $(\mathcal{Q}, \mathcal{G})$ two groups, the well-structured optimization problem matches with the framework of ADMM.

Next, we present the details of each subproblem. Since variables $\mathcal{Z}$, $\mathcal{E}$, and $\mathcal{M}$ are decoupled from each other, they can be solved separately.

1) *$\mathcal{Z}$-Subproblem:* The $\mathcal{Z}$-subproblem is

$$\arg\min_{\mathcal{Z}} \|\mathcal{Z}\|_{\mathrm{TTNN}} + \frac{\beta_1}{2}\|\mathcal{Z} - \mathcal{Q}^{(t)} + \frac{\mathcal{T}^{(t)}}{\beta_1}\|_F^2. \quad (14)$$

According to the proximal mapping of TTNN [54], the minimizer is given by

$$\mathcal{Z}^{(t+1)} = \mathcal{U}^{(t)} \diamond_\Phi \mathcal{S}_{\beta_1}^{(t)} \diamond_\Phi (\mathcal{V}^{(t)})^H \quad (15)$$

where $\mathcal{Q}^{(t)} - (\mathcal{T}^{(t)}/\beta_1) = \mathcal{U}^{(t)} \diamond_\Phi \mathcal{S}^{(t)} \diamond_\Phi (\mathcal{V}^{(t)})^H$, $\mathcal{S}_{\beta_1}^{(t)} = \Phi^H\{\max\{\bar{\mathcal{S}}_\Phi - (1/\beta_1), 0\}\}$.

2) *$\mathcal{E}$-Subproblem:* The $\mathcal{E}$-subproblem is

$$\arg\min_{\mathcal{E}} \lambda\|\mathcal{E}\|_1 + \frac{\beta_2}{2}\|\mathcal{E} - \mathcal{G}^{(t)} + \frac{\mathcal{H}^{(t)}}{\beta_2}\|_F^2. \quad (16)$$

It can be calculated by the soft shrinkage operator

$$\mathcal{E}^{(t+1)} = \max\left(|\tilde{\mathcal{E}}^{(t)}| - \frac{\lambda}{\beta_2}, 0\right) \circ (\tilde{\mathcal{E}}^{(t)}/|\tilde{\mathcal{E}}^{(t)}|) \quad (17)$$

where $\tilde{\mathcal{E}}^{(t)} = \mathcal{G}^{(t)} - \mathcal{H}^{(t)}/\beta_2$, $\circ$ is the componentwise multiplication, and the division is also performed componentwise.

3) *$\mathcal{M}$-Subproblem:* The $\mathcal{M}$-subproblem is

$$\arg\min_{\mathcal{M}} \frac{\beta_3}{2}\|\mathcal{M} - \mathcal{A} \diamond_\Phi \mathcal{Q}^{(t)} - \mathcal{G}^{(t)} + \frac{\mathcal{N}^{(t)}}{\beta_3}\|_F^2 + \Psi_\Omega(\mathcal{X} - \mathcal{M}). \quad (18)$$

Thus, we have

$$\mathcal{M}^{(t+1)} = P_{\Omega^c}\left(\mathcal{A} \diamond_\Phi \mathcal{Q}^{(t)} + \mathcal{G}^{(t)} - \frac{\mathcal{N}^{(t)}}{\beta_3}\right) + P_\Omega(\mathcal{X}) \quad (19)$$

where Ω^c is the orthogonal complement of Ω.

4) *$(\mathcal{Q}, \mathcal{G})$-Subproblem:* The $(\mathcal{Q}, \mathcal{G})$-subproblem is

$$\begin{aligned}\arg\min_{\mathcal{Q},\mathcal{G}} &\frac{\beta_3}{2}\left\|\mathcal{M}^{(t+1)} - \mathcal{A} \diamond_\Phi \mathcal{Q} - \mathcal{G} + \frac{\mathcal{N}^{(t)}}{\beta_3}\right\|_F^2 \\ &+ \frac{\beta_1}{2}\left\|\mathcal{Z}^{(t+1)} - \mathcal{Q} + \frac{\mathcal{T}^{(t)}}{\beta_1}\right\|_F^2 \\ &+ \frac{\beta_2}{2}\left\|\mathcal{E}^{(t+1)} - \mathcal{G} + \frac{\mathcal{H}^{(t)}}{\beta_2}\right\|_F^2.\end{aligned}$$

The solutions satisfy the following equations:

$$\begin{cases}\left(\dfrac{\beta_1}{\beta_3}\mathcal{I} + \dfrac{\beta_2}{\beta_2+\beta_3}\mathcal{A}^H \diamond_\Phi \mathcal{A}\right)\mathcal{Q}^{(t+1)} \\ \qquad = \dfrac{\mathcal{J}_1}{\beta_3} - \dfrac{1}{\beta_2+\beta_3}\mathcal{A}^H \diamond_\Phi \mathcal{J}_2 \\ \mathcal{G}^{(t+1)} = \dfrac{\mathcal{J}_2}{\beta_2+\beta_3} - \dfrac{\beta_3}{\beta_2+\beta_3}\mathcal{A} \diamond_\Phi \mathcal{Q}^{(t+1)}\end{cases} \quad (20)$$

where $\mathcal{J}_1 = \beta_1\mathcal{Z}^{(t+1)} + \mathcal{T}^{(t)} + \beta_3\mathcal{A}^H \diamond_\Phi (\mathcal{M}^{(t+1)} + (\mathcal{N}^{(t)}/\beta_3))$ and $\mathcal{J}_2 = \beta_2\mathcal{E}^{(t+1)} + \mathcal{H}^{(t)} + \beta_3\mathcal{M}^{(t+1)} + \mathcal{N}^{(t)}$.

After obtaining the representation tensor $\mathcal{Z}$, we set the affinity matrix by

$$\hat{\mathbf{Z}} = \frac{1}{2n_3}\sum_{i=1}^{n_3}(|\mathcal{Z}^{(i)}| + |(\mathcal{Z}^{(i)})^H|) \quad (21)$$

Algorithm 3 ADMM for Solving the Proposed Model (10)

Input: the observed tensor $\mathcal{X}$, dictionary $\mathcal{A}$, parameters λ, β_1, β_2, β_3, $\rho = 1.2$, and $\beta_{max} = 10^{10}$.

Output: the recovery and clustering results.

1: Initializations $\mathcal{M}^{(0)}$, $\mathcal{Z}^{(0)}$, $\mathcal{E}^{(0)}$, $\mathcal{Q}^{(0)}$, $\mathcal{G}^{(0)}$, maximum iterations $k_{\max} = 500$.
2: **While** $\frac{\|\mathcal{Z}^{(t+1)} - \mathcal{Z}^{(t)}\|_F}{\|\mathcal{Z}^{(t)}\|_F} > 10^{-4}$ and $k \le k_{\max}$ **Do**
3: Updating $\mathcal{Z}$ via (15);
4: Updating $\mathcal{E}$ via (17);
5: Updating $\mathcal{M}$ via (19);
6: Updating $\mathcal{Q}$ and $\mathcal{G}$ via (20);
7: Updating multipliers with

$$\begin{aligned}\mathcal{T}^{(t+1)} &= \mathcal{T}^{(t)} + \beta_1(\mathcal{Z}^{(t+1)} - \mathcal{Q}^{(t+1)}),\\ \mathcal{H}^{(t+1)} &= \mathcal{H}^{(t)} + \beta_2(\mathcal{E}^{(t+1)} - \mathcal{G}^{(t+1)}),\\ \mathcal{N}^{(t+1)} &= \mathcal{N}^{(t)} + \beta_3(\mathcal{M}^{(t+1)} - \mathcal{A} \diamond_\Phi \mathcal{Q}^{(t+1)} - \mathcal{G}^{(t+1)});\end{aligned}$$

8: $\beta_l^{(t+1)} = \min(\rho\beta_l^{(t)}, \beta_{max})$, $l = 1, 2, 3$;
9: **End Do**
10: Obtaining $\hat{\mathbf{Z}}$ by (21) and recovery $\hat{\mathcal{L}} = \mathcal{A} \diamond_\Phi \mathcal{Z}^{(t+1)}$;
11: Applying the spectral clustering method (Ncut) with the affinity matrix $\hat{\mathbf{Z}}$;
12: **Return**: the recovery and clustering results.

where $\mathcal{Z}^{(i)} \in \mathbb{R}^{n_2 \times n_2}$ represents the frontal slices of $\mathcal{Z}$. Then, we apply the spectral clustering approach NCut [58] to obtain the clustering result. Finally, we summarize the proposed algorithm in Algorithm 3. The total space cost of our algorithm is $\mathcal{O}(n_2n_3(n_1 + n_2))$.

Due to the convexity of the proposed model, the convergence of ADMM is guaranteed [59], [60], [61]. Accordingly, the proposed algorithm can converge to an optimal solution.

Complexity Analysis: The proposed algorithm contains four subproblems: $\mathcal{Z}$-subproblem, $\mathcal{E}$-subproblem, $\mathcal{M}$-subproblem, and $(\mathcal{Q}, \mathcal{G})$-subproblem. At each iteration, the major cost of updating $\mathcal{Z}^{(t+1)}$ involves the n_3 SVD on $n_2 \times n_2$ matrices at the cost of $\mathcal{O}(n_2^3n_3)$ and the unitary transform and its inverse operator on $n_2 \times n_2 \times n_3$ tensors at the cost of $\mathcal{O}(n_2^2n_3^2)$. When updating $\mathcal{E}^{(t+1)}$, the computational cost is $\mathcal{O}(n_1n_2n_3)$. The major computation of updating $\mathcal{M}^{(t+1)}$ is the Φ-product $\mathcal{A} \diamond_\Phi \mathcal{Q}^{(t)}$ at the cost of $\mathcal{O}(n_2n_3(n_1n_2 + n_1n_3 + n_2n_3))$. When updating $(\mathcal{Q}^{(t+1)}, \mathcal{G}^{(t+1)})$, the major cost is to compute the Φ-product at the cost of $\mathcal{O}(n_2n_3(n_1n_2 + n_1n_3 + n_2n_3))$. Therefore, the cost of Algorithm 3 at each iteration is $\mathcal{O}(n_2n_3(n_2^2 + n_1n_2 + n_1n_3 + n_2n_3))$.

VI. Tensor Data Recovery

We test the performance of the proposed method on tensor data recovery, including videos and faces data. For comparison, we choose seven state-of-the-art methods: HaLRTC [45], TNN [47], TRPCA [50], TLRR [33], robust tensor completion via tensor-train (RTC-TT) [14], Fourier transform-based transformed TNN (FTTNN) [57], and unitary transform-based transformed TNN (UTTNN) [54]. We scale the pixel value of test data to the interval [0, 1]. All experiments are performed in MATLAB R2018b with 32 GB of RAM and an Intel Core i7-8700M CPU with 3.70 GHz on a desktop.

We use two different unitary transforms in the proposed method. The first one is the discrete cosine transform (termed TTLRR-DCT). The second one is the unitary transform learned by each given data (termed as TTLRR-Data). Particularly, we first obtain an estimator $\mathcal{L}$ by FTTNN and unfold $\mathcal{L}$ along the third mode into a matrix $\boldsymbol{L}_{(3)}$. Then, we compute the SVD of $\boldsymbol{L}_{(3)}$ as $\boldsymbol{L}_{(3)} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^H$. Finally, we use the desired unitary matrix $\boldsymbol{U}^H$ [54] in T-TSVD for data recovery and clustering.

The choice of dictionary $\mathcal{A}$ is important for TTLRR. Since the observation data are incomplete with sparse errors, we use the estimated data by UTTNN [54] as the dictionary $\mathcal{A}$ in the data recovery task.

Evaluation Metrics: We use the peak signal-to-noise ratio (PSNR) and the structural similarity index (SSIM) [62] to measure the recovery performance of all methods. Specifically, PSNR is used to estimate the error between corresponding pixels, and SSIM is used to measure the similarity between two images. Higher PSNR and SSIM values reflect better data quality.

Competitors and Parameters Settings: The proposed method involves the following parameters: λ, β_1, β_2, and β_3. We set $\lambda = (a/(\rho n_{(1)} n_3)^{1/2})$ and empirically choose a from $\{5, 10, 20, 30, 40, 50\}$. For penalties β_1, β_2, and β_3, we empirically choose them from $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$.

For all competing methods, the parameters are optimally assigned or tuned as suggested in the corresponding papers for obtaining the highest PSNR value.

1) *HaLRTC [45]:* This method focused on tensor completion by SNN. The weighted parameters α_i $(i = 1, 2, 3)$ are set to $\alpha_i = 1/3$ $(i = 1, 2, 3)$. The penalty parameter β can be selected from $\{5 \times 10^{-5}, 10^{-4}, 5 \times 10^{-4}, 10^{-3}, 5 \times 10^{-3}\}$.
2) *TNN [47]:* This method minimized the TNN to recover the third-order tensor from missing pixels. There is only a penalty parameter β. We empirically choose the penalty parameter $\beta \in \{10^{-6}, 5 \times 10^{-6}, 10^{-5}, 5 \times 10^{-5}, 10^{-4}\}$ for the highest PSNR value.
3) *TRPCA [50]:* This method focused on exactly recovering the low-rank component from noisy data. According to [50, Th. 3.1], the regularization parameter takes $1/(\max(n_1, n_2)n_3)^{1/2}$ in all experiments.
4) *TLRR [33]:* The method proposed a tensor low-rank representation method to recover the clean data from the noisy observation. We set regularization parameter to $a/(\max(n_1, n_2)n_3)^{1/2}$, and a is selected from $\{0.001, 0.005, 0.01, 0.1, 1\}$ to obtain the good performance. Moreover, this method applies the estimated image by UTTNN [54] as the dictionary.
5) *RTC-TT [14]:* This method solved the robust tensor completion problem by tensor train rank. In this method, an autoweighted mechanism is used to guide the importance of different modes of the tensor. The regularization parameter is set to $1/(\max(n_1, n_2)n_3)^{1/2}$.
6) *FTTNN [57]:* In the Fourier transform, this method recovered a low-rank tensor from an undersampled and possibly arbitrarily corrupted measurement. The regularization parameter is set as $a/(\max(n_1, n_2)n_3)^{1/2}$, and a is selected from $\{1.1, 1.3, 1.5, 1.7, 2\}$.
7) *UTTNN [54]:* Using the unitary transform learned from the given data, this method studied robust tensor completion by the transformed tensor SVD. The regularization parameter is set as $a/(\max(n_1, n_2)n_3)^{1/2}$ and a is selected from $\{10, 15, 18, 20, 23, 25, 28, 30, 33, 35, 40, 45, 50\}$. The penalty parameter β is chosen from $\{0.001, 0.005, 0.01, 0.05, 0.1\}$.

TABLE I

EXACT RECOVERY PERFORMANCE ($\|\hat{\mathcal{L}} - \mathcal{L}\|_F / \|\mathcal{L}\|_F$) OF THE SYNTHETIC DATA WITH DIFFERENT SIZES

$n_1 \times n_2 \times n_3$	ρ	γ	UTTNN	TTLRR-Data
$100 \times 50 \times 100$	0.9	0.05	1.05×10^{-2}	1.79×10^{-4}
		0.1	1.07×10^{-2}	5.99×10^{-4}
	0.8	0.05	1.06×10^{-2}	5.99×10^{-4}
		0.1	1.95×10^{-2}	4.33×10^{-2}
$200 \times 100 \times 200$	0.9	0.05	5.26×10^{-3}	9.95×10^{-4}
		0.1	5.27×10^{-3}	1.30×10^{-3}
	0.8	0.05	1.01×10^{-2}	1.20×10^{-3}
		0.1	1.12×10^{-2}	5.89×10^{-2}

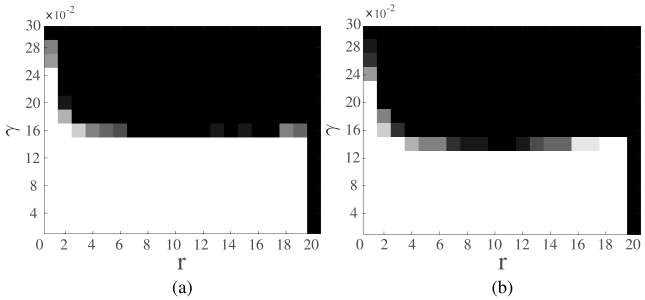


Fig. 4. Exact recovery of TTLRR-Data for varying rank and noise corruption (the white region indicates the success case and the black region indicates the failure case). (a) SR = 0.95. (b) SR = 0.90.

A. Synthetic Data Recovery

Here, we evaluate the recovery guarantee in Theorem 1 on our TTLRR-Data using randomly synthetic data.

1) Exact Recovery on Random Data With Different Sizes: We generate the random tensor $\mathcal{L} = [\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_5] \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ with $\mathcal{L}_q = \mathcal{C}_q * \mathcal{D}_q$ $(q = 1, 2, \ldots, 5)$, where the entries of $\mathcal{C}_q \in \mathbb{R}^{n_1 \times p \times n_3}$ and $\mathcal{D}_q \in \mathbb{R}^{p \times s \times n_3}$ are randomly sampled from i.i.d. $\mathcal{N}(0, 1)$. Here, we set the tensor size as $n_1 = n_3 = 100$ and $p = s = 10$ and $n_1 = n_3 = 200$ and $p = s = 20$. So the numbers n_2 of two tensors are $n_2 = 50$ and $n_2 = 100$, respectively. In addition, the entries are independently corrupted with probability γ to generate the noisy tensor. Then, the observed tensor is obtained by choosing a percentage ρ entries of the noisy tensor. Table I reports the recovery results of different sampling rates (ρ) and noise rates (γ). One can see that the relative errors of ($\|\hat{\mathcal{L}} - \mathcal{L}\|_F / \|\mathcal{L}\|_F$) of our method are very small. Meanwhile, the proposed method achieves smaller relative errors than those obtained UTTNN in most cases. This confirms the exact recovery performance of the proposed method, as shown in Theorem 1, and the advantage of our method.

2) Exact Recovery on Varying Rank and Sparsity: Now, we examine the recovery performance of our method with varying tubal rank $r := \text{rank}_t(\mathcal{L})$ and varying noise rate γ. Similar to the above-mentioned generation process, we produce the random tensor $\mathcal{L}_q = \mathcal{C}_q * \mathcal{D}_q$ ($\mathcal{C}_q \in \mathbb{R}^{n_1 \times r_q \times n_3}$ and $\mathcal{D}_q \in \mathbb{R}^{r_q \times s_q \times n_3}$, $q = 1, 2, \ldots, 5$) and set the tensor $\mathcal{L} = [\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_5] \in \mathbb{R}^{n_1 \times n_2 \times n_3}$. Hence, $\mathcal{L}$ contains $n_2 = \sum_{q=1}^{5} s_q$ samples with tubal rank $r = \sum_{q=1}^{5} r_q$. We set $n_1 = 200$, $n_3 = 20$, $s_q = s = 20$, $\gamma = [0.02 : 0.02 : 0.3]$, and the rank $r = [1 : 20]$. For each pair (r, γ), we perform ten

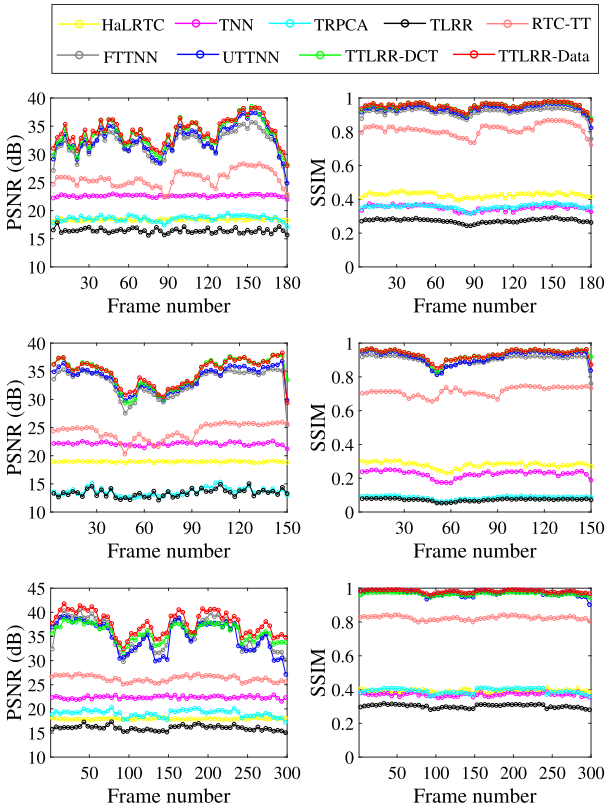


Fig. 5. Performance (PSNR and SSIM) of different methods on video datasets for each frame with SR = 0.6 and NR = 0.1. From top to bottom: *Carphone*, *Suzie*, and *News*.

trials and declare a trial to be successful if the recovered $\hat{\mathcal{L}}$ satisfies $(\|\hat{\mathcal{L}} - \mathcal{L}\|_F / \|\mathcal{L}\|_F) \leq 5 \times 10^{-3}$. Fig. 4 shows the experimental results of TTLRR-Data for different sampling rates, i.e., $\rho = 0.95$ and $\rho = 0.90$. One can see that in both cases, there exists a big region where TTLRR-Data can exactly recover the underlying data. This experiment also verifies the main result in Theorem 1.

B. Video Recovery

1) Settings: We test the performance of the proposed method on videos, including *Carphone* $144 \times 176 \times 180$, *Suzie* $144 \times 176 \times 150$, and *News* $144 \times 176 \times 300$[1] to validate the effectiveness of the proposed method in the robust tensor completion problem. We randomly generate incomplete data with different sampling rates (SR = 0.4, 0.6, and 0.8) and different noise rates (NR = 0.1 and 0.2).

2) Results: Table II shows the numerical results of the recovered videos by different methods with different SRs and NRs. The best and suboptimal results are denoted in bold fronts and underlined values, respectively. One can see that the proposed TTNN-Data achieves the highest PSNR and SSIM values in most cases. This suggests the superiority of the unitary transform learned from the given tensor data. Moreover, the PSNR values obtained by TTLRR-DCT and TTLRR-Data are higher than those obtained by FTTNN and UTTNN. This suggests that the original high-dimensional data depends on multiple subspaces instead of one space. Fig. 5 plots the PSNR and SSIM curves of the recovered results by a different method in terms of each frame of three videos. Obviously, the proposed method TTLRR-Data achieves the highest PSNR and SSIM values in most frames.

Fig. 6 presents the recovered videos of compared methods with SR = 0.6 and NR = 0.1. One can see that HaLRTC and TNN only recover the coarse structure of the videos while remaining lots of noise. TRPCA and TLRR methods only consider sparsely corrupted observations, so there are still missing pixels in the restored results. In comparison, both TTLRR-DCT and TTLRR-Data outperform the compared methods in recovering the details and structures.

C. Face Data Recovery

1) Settings: In this part, we test the performance of the proposed method on the face image extended Yale B (YaleB) database [63]. The YaleB contains 38 subjects and each subject contains 64 images under various lighting conditions. In this experiment, the dataset contains 840 frontal face images of 20 subjects and each image is resized to 80×60. We randomly sample the YaleB dataset and add sparse noise to generate the observed data. The SRs are set as 0.4, 0.6, and 0.8, and sparse NRs are set as 0.1 and 0.2, respectively.

2) Results: Table III shows the PSNR and SSIM values obtained by different methods. It can be seen that the PSNR and SSIM values obtained by TTLRR-DCT and TTLRR-Data are higher than those obtained by other compared methods in most cases.

Fig. 7 presents the recovered face images by different methods with SR = 0.6 and NR = 0.2. Obviously, TTLRR-DCT and TTLRR-Data produce more satisfactory recovered images than other compared methods. We also observe that the recovered images of TTLRR-Data are clear than those obtained by TTLRR-DCT. TTLRR-Data can preserve more details and distinguishable features than the other compared methods, which can be beneficial to enhance the performance of the downstream clustering tasks.

VII. Tensor Data Clustering

In this section, we conduct various experiments on face data, object images, and scene data to test the effectiveness of the proposed methods (TTLRR-DCT and TTLRR-Data) for data clustering. Moreover, we compare the proposed methods with several relevant and state-of-the-art approaches, namely, low rank subspace clustering (LRSC) [64], latent multi-view subspace clustering (LMSC) [65], RSS [66], robust kernel low-rank representation (RKLRR) [67], implicit block diagonal low-rank representation (IBDLR) [68], stochastic sparse subspace clustering via orthogonal matching pursuit (S3COMP) [69], TLRR [33], block diagonal representation (BDR) [70], and generalized nonconvex low-rank tensor approximation (GNLTA) [35]. Among them, IBDLR and BDR incorporate the block diagonal prior to guaranteeing the block diagonal of the affinity matrix. All test data are scaled into the interval [0, 1]. In the following, we use the estimated data by FTTNN [57] as dictionary $\mathcal{A}$.

Evaluation Metrics: For the quantitative measures of clustering results, we utilize the unsupervised clustering accuracy (ACC) [24], the normalized mutual information (NMI) [71], and the purity (PUR) [72]. ACC is the accuracy metric computed for the best matching permutation between clustered labels and ground-truth labels. NMI is an information theoretic-based measure and can balance the quality of the

[1] http://trace.eas.asu.edu/yuv/

TABLE II

PERFORMANCE (PSNR AND SSIM) OF TENSOR DATA RECOVERY ON THREE VIDEO DATASETS WITH DIFFERENT SRS AND NRS. THE **BEST** VALUES ARE HIGHLIGHTED BY BOLD FRONTS AND THE **SUBOPTIMAL** VALUES ARE HIGHLIGHTED BY UNDERLINED

Dataset	SR	NR	Methods	HaLRTC	TNN	TRPCA	TLRR	RTC-TT	FTTNN	UTTNN	TTLRR-DCT	TTLRR-Data
Carphone	0.4	0.1	PSNR	19.11	23.71	6.86	12.71	21.52	30.01	30.60	31.72	**31.86**
			SSIM	0.4426	0.3946	0.0786	0.1725	0.5285	0.9048	0.9205	0.9324	**0.9332**
		0.2	PSNR	16.52	20.91	13.76	13.91	17.29	27.65	29.00	30.38	**30.60**
			SSIM	0.3133	0.2685	0.0654	0.0715	0.3442	0.8029	0.8638	**0.9032**	0.9031
	0.6	0.1	PSNR	18.42	22.66	18.62	21.63	25.61	31.85	32.99	33.57	**33.93**
			SSIM	0.4279	0.3516	0.3558	0.6794	0.8137	0.9174	0.9409	0.9523	**0.9539**
		0.2	PSNR	15.67	19.81	18.47	20.02	24.37	25.70	30.77	30.90	**31.76**
			SSIM	0.3052	0.2315	0.3218	0.5859	0.7152	0.6760	0.9040	0.9019	**0.9098**
	0.8	0.1	PSNR	17.67	21.72	30.23	33.77	27.33	32.90	34.70	35.10	**35.56**
			SSIM	0.4172	0.3162	0.9266	0.9514	0.8695	0.9188	0.9498	0.9643	**0.9660**
		0.2	PSNR	14.86	18.81	30.07	30.57	26.66	22.38	31.60	31.10	**31.93**
			SSIM	0.3026	0.1986	0.9097	0.8481	0.8079	0.5391	0.9081	0.8905	**0.9196**
Suzie	0.4	0.1	PSNR	19.84	23.16	13.90	13.91	22.02	28.28	27.48	**29.92**	29.44
			SSIM	0.3242	0.2737	0.0261	0.0272	0.6171	0.8409	0.8194	**0.8589**	0.8478
		0.2	PSNR	17.20	20.41	13.65	13.75	22.33	28.50	29.86	30.69	**30.86**
			SSIM	0.1857	0.1487	0.0217	0.0221	0.6135	0.7414	0.8555	0.8702	**0.8714**
	0.6	0.1	PSNR	18.94	22.09	13.55	20.43	25.94	33.05	33.56	34.91	**34.96**
			SSIM	0.2804	0.2263	0.0857	0.4841	0.7577	0.9004	0.9065	**0.9368**	0.9357
		0.2	PSNR	16.14	19.32	14.67	26.08	24.63	29.58	31.10	32.40	**32.52**
			SSIM	0.1557	0.1189	0.0303	0.7599	0.7156	0.8743	0.8504	0.8989	**0.8994**
	0.8	0.1	PSNR	18.07	21.21	31.45	33.37	27.79	34.09	34.86	36.52	**36.75**
			SSIM	0.2516	0.1977	0.9112	0.9140	0.7913	0.9043	0.9120	0.9531	**0.9533**
		0.2	PSNR	15.24	18.27	31.07	31.81	27.90	31.20	30.57	32.54	**33.08**
			SSIM	0.1404	0.0934	0.8908	0.8820	0.7752	**0.9064**	0.7964	0.8919	0.8975
News	0.4	0.1	PSNR	18.81	23.50	9.50	10.77	21.98	32.38	31.03	32.29	**34.97**
			SSIM	0.4256	0.4219	0.0710	0.1213	0.7195	0.9620	0.9453	0.9312	**0.9677**
		0.2	PSNR	16.17	20.71	9.52	10.93	21.32	31.11	30.73	33.08	**33.49**
			SSIM	0.2882	0.2893	0.0704	0.1175	0.6732	0.9297	0.9139	0.9537	**0.9580**
	0.6	0.1	PSNR	17.92	22.37	19.04	26.27	26.31	35.18	35.16	35.34	**37.77**
			SSIM	0.3909	0.3695	0.3915	0.8532	0.8233	0.8233	0.9752	0.9643	**0.9804**
		0.2	PSNR	15.14	19.36	18.94	27.11	26.04	28.80	32.10	33.16	**33.59**
			SSIM	0.2649	0.2313	0.3631	0.7647	0.8400	0.7885	0.9608	0.9563	**0.9610**
	0.8	0.1	PSNR	17.07	21.44	31.56	33.59	28.50	37.36	37.48	38.17	**40.19**
			SSIM	0.3642	0.3273	0.9632	0.9686	0.9183	0.9814	0.9793	0.9772	**0.9860**
		0.2	PSNR	14.26	18.43	31.48	33.36	26.48	31.71	33.67	35.32	**35.67**
			SSIM	0.2493	0.2004	0.9601	0.9573	0.7699	0.9597	0.9657	0.9707	**0.9732**

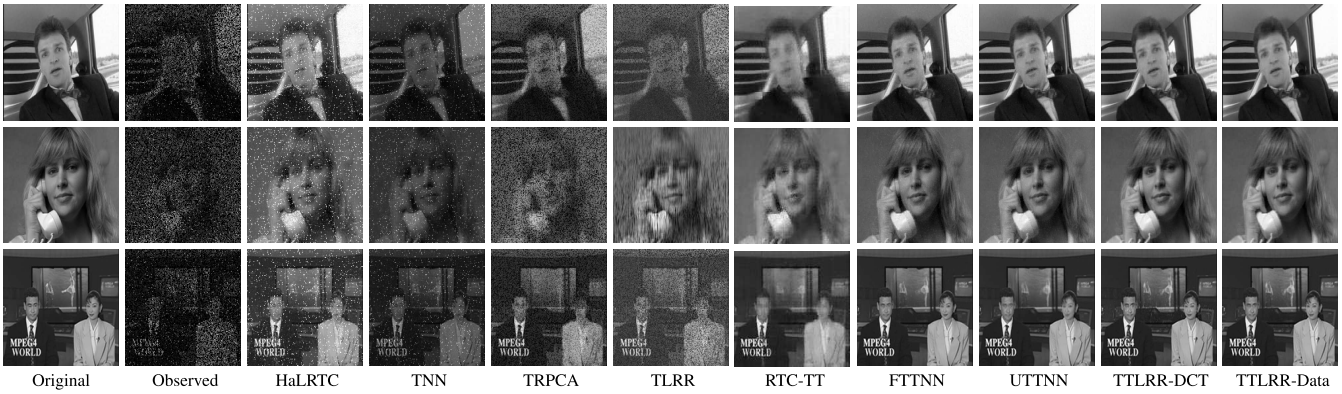


Fig. 6. Recovered videos by different methods with SR = 0.6 and NR = 0.1. From top to bottom: *Carphone* (50th frame), *Suzie* (100th frame), and *News* (tenth frame).

clustering against the number of clusters. PUR is measured by the percentage of the total number of objects that are correctly classified. Higher ACC, NMI, and PUR values indicate more accurate clustering results.

Competitors and Parameter Settings: In tensor data clustering, the parameter setting of the proposed method is as follows. We set $\lambda = (a/(\rho n_{(1)} n_3)^{1/2})$ and a is selected from $\{1, 2, 4, 5, 10, 20, 30, 40\}$. For penalties β_1, β_2, and β_3, we empirically choose them from $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$.

For the compared methods, we use the codes provided by the authors and tune the parameters according to the authors' suggestions in their papers to obtain the best result.

1) *LRSC [64]:* The method proposed a general low-rank matrix subspace clustering framework for solving the clustering problem in the case of data corrupted by noise and/or gross errors. There are parameters α and τ in LRSC (the P3 model). According to the authors' suggestion, we choose them from $\alpha \in \{0.03, 0.045, 0.07, 0.1\}$ and $\tau \in \{0.01, 0.03, 0.045, 0.07, 0.1\}$, respectively.
2) *LMSC [65]:* The method proposed a latent subspace clustering method, which contains the regularization

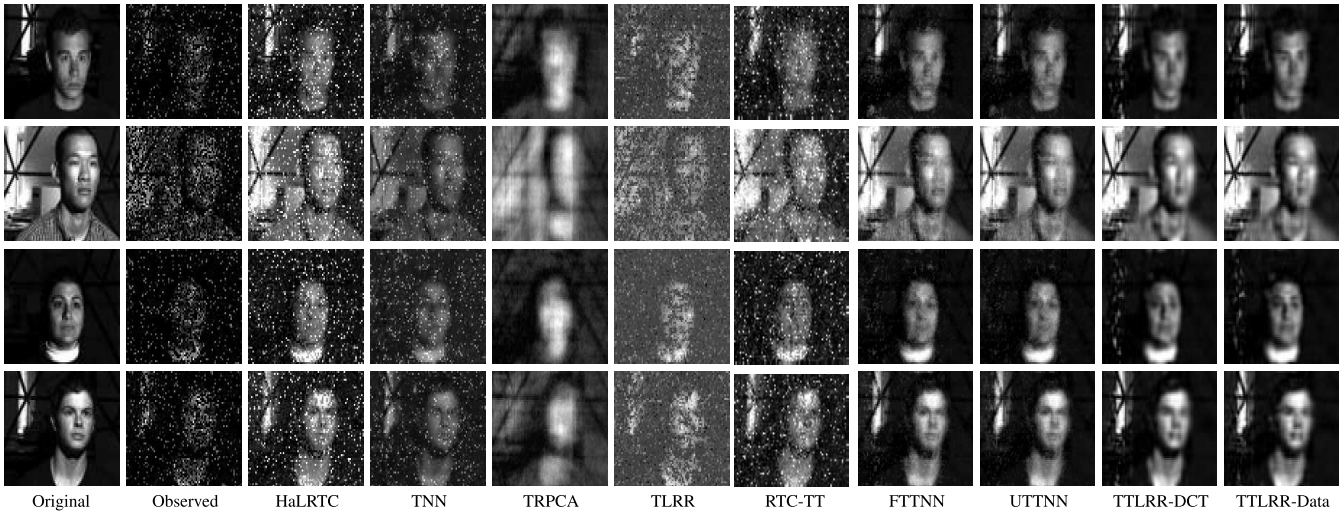


Fig. 7. Recovered faces by different methods with SR = 0.6 and NR = 0.2.

TABLE III

PERFORMANCE (PSNR AND SSIM) OF TENSOR DATA RECOVERY ON YALEB DATASET WITH DIFFERENT SRS AND NRS. THE **BEST** VALUES ARE HIGHLIGHTED BY BOLD FRONTS AND THE **SUBOPTIMAL** VALUES ARE HIGHLIGHTED BY UNDERLINED

SR	NR	Methods	HaLRTC	TNN	TRPCA	TLRR	RTC-TT	FTTNN	UTTNN	TTLRR-DCT	TTLRR-Data
0.4	0.1	PSNR	17.19	22.47	11.98	19.00	17.54	22.16	23.56	23.81	**23.91**
		SSIM	0.4250	0.4178	0.0592	0.2986	0.4348	**0.8389**	0.8329	0.8159	0.8109
	0.2	PSNR	14.79	20.20	11.98	18.31	17.07	21.85	22.13	21.87	**22.79**
		SSIM	0.3067	0.3159	0.0592	0.2186	0.3866	0.7893	**0.7989**	0.7917	0.7945
0.6	0.1	PSNR	16.72	22.13	13.93	20.92	18.86	25.05	25.65	26.11	**26.20**
		SSIM	0.4257	0.4225	0.4751	0.6835	0.4770	**0.8863**	0.8350	0.8707	0.8730
	0.2	PSNR	14.07	19.54	14.05	21.55	17.29	23.61	23.98	24.47	**24.76**
		SSIM	0.3085	0.3105	0.4884	0.6281	0.3969	0.7762	0.8037	0.8323	**0.8388**
0.8	0.1	PSNR	16.10	21.56	20.26	26.86	21.11	27.30	27.53	28.20	**28.29**
		SSIM	0.4189	0.4092	0.8274	0.8487	0.7239	0.8956	**0.9132**	0.9024	0.9057
	0.2	PSNR	13.34	18.79	20.43	24.01	18.14	22.83	23.91	25.97	**26.10**
		SSIM	0.3046	0.2930	0.8224	0.8422	0.4264	0.6826	0.7303	0.8601	**0.8626**

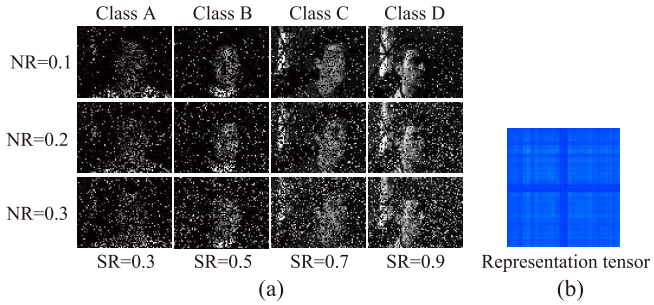


Fig. 8. (a) Examples of the corrupted face data from the YaleB datasets. (b) Block-diagonal structure of the observed data with SR = 0.5 and NR = 0.1. It shows the bad clustering performance from the corrupted data.

parameter λ and the penalty parameter μ. We tune the parameter λ from $\{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}\}$ suggested by authors and empirically set $\mu = 10^{-4}$ for the highest ACC value.

3) *RSS [66]:* The subspace clustering method can simultaneously learn the representations of data and the affinity matrix. There are three regularization parameters λ_1, λ_2, and λ_3 in RSS. According to the suggestion in RSS, we set $\lambda_1 = \lambda_2 = \lambda_3 = \hat{\lambda} \in \{0.1, 0.2, \ldots, 1.0\}$.
4) *RKLRR [67]:* The method showed a RKLRR to deal with the nonlinear corrupted data. The method only has one tradeoff parameter λ that can be selected from $\{0.1, 0.2, 0.3, 0.5\}$ to obtain the best performance.
5) *IBDLR [68]:* The method incorporated the block diagonal prior into the low-rank representation method for data clustering. There are two regularization parameters λ and γ. We empirically choose them from $\{1, 3, 5, 7, 9, 11\}$ for the highest ACC value.
6) *S3COMP [69]:* This is a scalable and flexible sparse subspace clustering approach. It mainly contains three parameters λ, δ, and T. According to the authors' suggestion, we select λ from $\{0.01, 0.02, \ldots, 0.09\}$, δ is selected from $\{0.1, 0.2, \ldots, 0.9\}$, and T is fixed to 15.
7) *TLRR [33]:* The method used the tensor low-rank representation method based on TNN to cluster the corrupted data. We choose the Robust TLRR as the comparison method and set the regularization parameter to $a/\sqrt{\max(n_1, n_2)n_3}$, where a is selected from $\{0.05, 0.1, 1, 5, 10, 20\}$ to obtain the best performance. Similar to our method, the estimated data by FTTNN [57] is used as the dictionary in their algorithm.
8) *BDR [70]:* This method is a matrix-based subspace clustering method and introduces the block diagonal structure prior. As suggested by the authors, the parameters λ and γ are, respectively, selected from $\{10, 20, 30, 40, 50, 60, 70\}$ and $\{0.001, 0.01, 0.1, 0.5, 1, 2, 3, 5, 10, 50\}$ in the numerical experiments.
9) *GNLTA [35]:* This method used the nonconvex function to approximate the low-rank representation tensor for data clustering. In the following experiments, we choose the nonconvex function Laplace to test the performance of GNLTA. Parameters α and θ are selected from interval $[0.005, 0.06]$ and $[1.1, 2]$, respectively.

A. Face Data Clustering

1) Settings: We test the performance of the proposed method on the YaleB dataset [63]. The YaleB has 2432 facial images of 38 subjects under various lighting conditions, where each subject contains 64 images. In our experiments, the dataset contains 840 frontal face images with the size of 80×60 of 20 subjects. We randomly sample the YaleB dataset and add sparse noise to generate the observed data. The SRs are set as 0.1–0.9 and sparse NRs are set as 0.1–0.3, respectively. The goal of clustering is to group the corrupted face data into different groups according to the subject. By arranging each data as the lateral slices of a third-order tensor, we perform the proposed subspace clustering

TABLE IV

PERFORMANCE (ACC, NMI, AND PUR) OF TENSOR DATA CLUSTERING ON THREE DATASETS WITH DIFFERENT SRS AND NRS. THE **BEST** VALUES ARE HIGHLIGHTED BY BOLD FRONTS AND THE **SUBOPTIMAL** VALUES ARE HIGHLIGHTED BY UNDERLINED

Dataset	SR	Methods	LRSC	LMSC	RSS	RKLRR	IBDLR	S3COMP	TLRR	BDR	GNLTA	TTNN-DCT	TTNN-Data
YaleB (NR = 0.2)	0.3	ACC	0.1331	0.1508	0.1107	0.0976	0.4758	0.1811	0.1292	0.1657	0.3498	0.7679	**0.8480**
		NMI	0.1869	0.2184	0.1382	0.1201	0.6133	0.2606	0.1840	0.2268	0.5136	0.8313	**0.9117**
		PUR	0.1413	0.1651	0.1173	0.1023	0.4968	0.1935	0.1370	0.1764	0.3751	0.7868	**0.8646**
	0.5	ACC	0.1373	0.1513	0.1196	0.0967	0.6549	0.3210	0.6783	0.1908	0.4171	0.8037	**0.8573**
		NMI	0.1957	0.2186	0.1461	0.1203	0.7723	0.3972	0.7672	0.2528	0.6103	0.8759	**0.9189**
		PUR	0.1462	0.1673	0.1269	0.1007	0.6773	0.3340	0.6965	0.2026	0.4420	0.8187	**0.8757**
	0.7	ACC	0.1477	0.1502	0.1244	0.0974	0.5925	0.5115	0.7711	0.2410	0.5651	0.8190	**0.8836**
		NMI	0.2102	0.2201	0.1510	0.1259	0.7456	0.6372	0.8397	0.3067	0.7364	0.8811	**0.9311**
		PUR	0.1588	0.1658	0.1435	0.1018	0.6143	0.5283	0.7896	0.2549	0.5963	0.8321	**0.8935**
COIL20 (NR = 0.1)	0.3	ACC	0.1326	0.2347	0.0972	0.1194	0.3962	0.1608	0.2800	0.1903	0.4803	0.6616	**0.7066**
		NMI	0.1098	0.3366	0.0485	0.0779	0.4488	0.1426	0.2852	0.1866	0.5699	0.7226	**0.7691**
		PUR	0.1419	0.2511	0.1016	0.1268	0.4147	0.1683	0.2972	0.2022	0.5033	0.6894	**0.7367**
	0.5	ACC	0.1233	0.2404	0.1031	0.1046	0.6508	0.2098	0.4508	0.1885	0.6545	0.6756	**0.7285**
		NMI	0.0899	0.3568	0.0456	0.0531	0.7227	0.2302	0.4790	0.1737	0.7411	0.7446	**0.7920**
		PUR	0.1307	0.2716	0.1107	0.1096	0.6623	0.2288	0.4822	0.1956	0.6714	0.7027	**0.7501**
	0.7	ACC	0.1194	0.2780	0.0962	0.0998	0.6435	0.3274	0.6047	0.1767	0.6539	0.6909	**0.7082**
		NMI	0.0799	0.3903	0.0439	0.0475	0.7421	0.3872	0.6807	0.1580	0.7586	0.7753	**0.7930**
		PUR	0.1263	0.3122	0.1010	0.1040	0.6590	0.3422	0.6474	0.1858	0.6735	0.7215	**0.7311**
UCSD (NR = 0.3)	0.3	ACC	0.1070	0.1398	0.1074	0.1041	0.2433	0.1180	0.1734	0.1179	0.3943	0.7556	**0.7890**
		NMI	0.0483	0.0956	0.0499	0.0445	0.2032	0.0615	0.1429	0.0609	0.4691	0.7600	**0.8177**
		PUR	0.1332	0.1578	0.1337	0.1309	0.2863	0.1484	0.2248	0.1518	0.4579	0.8042	**0.8305**
	0.5	ACC	0.1079	0.1382	0.1068	0.1073	0.5277	0.1403	0.5972	0.1123	0.6628	0.8099	**0.8293**
		NMI	0.0478	0.0950	0.0446	0.0482	0.5469	0.0807	0.6122	0.0502	0.7032	0.8194	**0.8402**
		PUR	0.1328	0.1564	0.1262	0.1351	0.5773	0.1770	0.6477	0.1423	0.7187	0.8534	**0.8731**
	0.7	ACC	0.1092	0.1403	0.1096	0.1055	0.6358	0.2088	0.5266	0.1215	0.7815	0.8246	**0.8497**
		NMI	0.0473	0.0953	0.0480	0.0477	0.6776	0.1735	0.5549	0.0630	0.8728	0.8597	**0.8866**
		PUR	0.1356	0.1570	0.1333	0.1299	0.6911	0.2457	0.5892	0.1524	0.8612	0.8725	**0.8962**

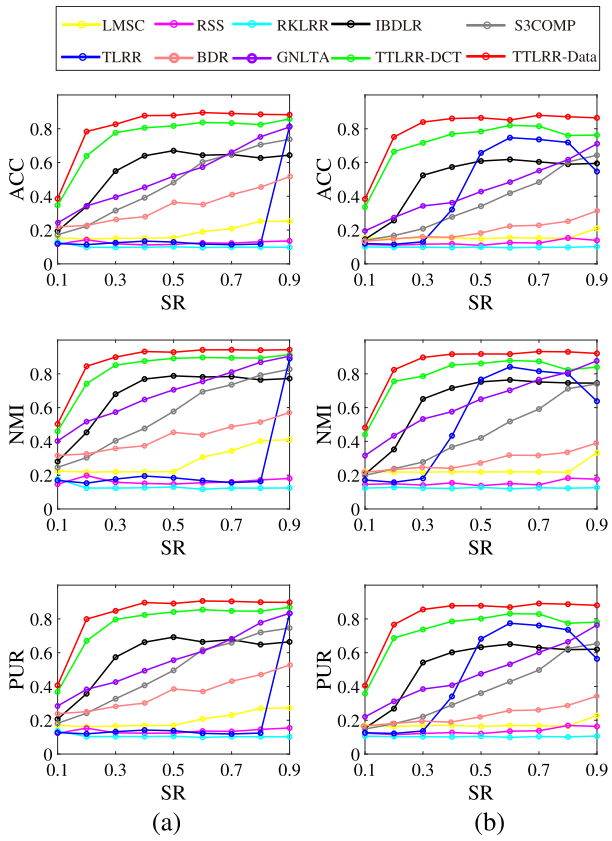


Fig. 9. Performance (ACC, NMI, and PUR) of different methods on YaleB dataset with different SRs. (a) NR = 0.1. (b) NR = 0.2.

method on the tensor. Fig. 8(a) shows some example data. Fig. 8(b) presents the block-diagonal structure of corrupted data and implies a bad cluster performance of directly clustering corrupted data.

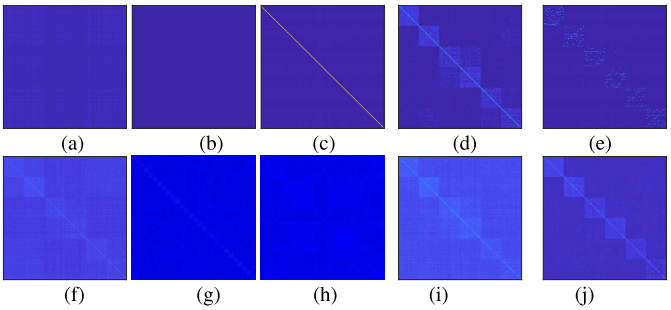


Fig. 10. Comparison of block-diagonal structures learned by different methods on YaleB dataset with NR = 0.2 and SR = 0.7. (a) LMSC. (b) RSS. (c) RKLRR. (d) IBDLR. (e) S3COMP. (f) TLRR. (g) BDR. (h) GNLTA. (i) TTLRR-DCT. (j) TTLRR-Data.

2) Results: Fig. 9 plots the ACC, NMI, and PUR curves against SR with NR $= 0.1, 0.2$, respectively. One can see that TTLRR-DCT and TTLRR-Data achieve the better performance than the compared methods in most cases. Table IV shows the clustering results (ACC, NMI, and PUR) by different subspace clustering methods with NR $= 0.2$ and SR $= 0.3, 0.5, 0.7$, respectively. One can see that the proposed method outperforms other compared methods. This suggests the effectiveness of the proposed method for clustering the corrupted data. Meanwhile, the improvement of the proposed method is more significant with the increased SR. By introducing sparse noise, the robustness of the proposed method is also verified.

Furthermore, Fig. 10 displays the block-diagonal structures of the affinity matrix obtained by different methods. For convenience, we only show the coefficients among six classes for observation. For methods TLRR, TTLRR-DCT, and TTLRR-Data, we plot $\hat{\boldsymbol{Z}}$ defined in (21) to show the block-diagonal structures. It can be seen that even without introducing the block-diagonal prior to IBDLR, the block-diagonal structure

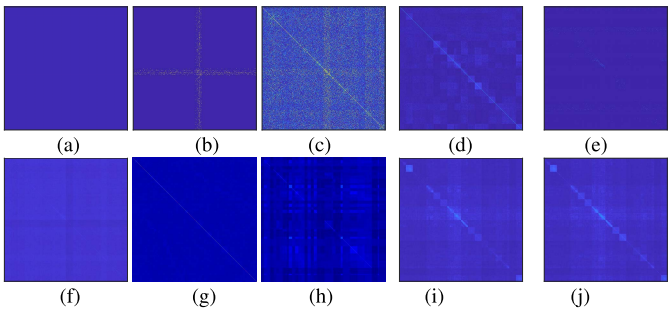


Fig. 11. Comparison of block-diagonal structures learned by different methods on UCSD dataset with NR = 0.3, SR = 0.7. (IBDLR has the obvious group effect outside the diagonal). (a) LMSC. (b) RSS. (c) RKLRR. (d) IBDLR. (e) S3COMP. (f) TLRR. (g) BDR. (h) GNLTA. (i) TTLRR-DCT. (j) TTLRR-Data.

learned by TTLRR-Data is clearer than other competitors. This demonstrates that the proposed model can approximately recover the tensor subspace of corrupted data, and thus, cluster data accurately.

B. Object Data Clustering

1) Settings: Next, we use the Columbia Object Image Library (COIL20) dataset[2] to test the performance of TTNN. The COIL20 contains 1440 images of 20 subjects including various objects, where each subject contains 74 images. And we resize each image to 32×32. The randomly sampling rates are set as 0.3, 0.5, and 0.7, and the sparse noise rate is set as 0.1, respectively. Corrupted data are put into a third-order tensor to evaluate the performance of the proposed method.

2) Results: In the middle rows of Table IV, we show the clustering performance of all methods on the COIL20 dataset. Similar to previous results, the proposed method achieves the best in preserving the intrinsic structures of data and clustering tasks. Meanwhile, it also exhibits the best robustness of our method in missing samples and noise situations.

C. Scene Data Clustering

1) Settings: The University of California at San Diego and Academic Science (USCD) dataset[3] contains 1200 different environment images from 18 video sequences. Similar to the COIL20 dataset, each image is normalized to 32×32. The randomly sampling rates are set as 0.3, 0.5, and 0.7. In order to show the robustness of the proposed method, we set the sparse noise as NR = 0.3. Next, we use the USCD dataset to evaluate the proposed method.

2) Results: In the bottom rows of Table IV, we present the clustering performance of all methods on the USCD dataset. Clearly, TTLRR-Data achieves the highest evaluation metrics. Moreover, the proposed method reaches a consistent significance over the other competitors when the sparse noise is high (NR = 0.3). This suggests the robustness of the proposed method.

Fig. 11 displays the block-diagonal structures of the affinity matrix obtained by different methods on the UCSD database. Although IBDLR employs the block-diagonal prior constraint to model the block-diagonal structure, it also has the obvious group effect outside the diagonal. This leads to lower clustering evaluation values, as shown in Table IV. Again, the proposed TTLRR-Data admits a better performance over other methods, which indicates the promising superiority of the unitary transform learned from the given data.

[2]http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php

[3]http://www.svcl.ucsd.edu/projects/backgroundsubtraction/ucsdbgsubdataset.htm

TABLE V

RUNNING TIME (IN SECONDS) OF TENSOR DATA RECOVERY ON CARPHONE AND NEWS DATASETS WITH DIFFERENT SRS AND NRS

Dataset	SR	HaLRTC	TNN	TRPCA	TLRR	RTC-TT	FTTNN	UTTNN	TTNN-DCT	TTNN-Data
Carphone (NR = 0.1)	0.4	9.91	1682.76	71.44	215.99	1659.33	42.62	343.02	129.36	117.39
	0.6	8.56	1671.85	70.01	233.95	1307.90	43.02	336.15	118.55	118.89
	0.8	9.23	1494.14	97.12	273.83	4310.24	43.28	331.44	128.47	119.78
News (NR = 0.2)	0.4	37.99	4269.40	107.74	315.73	2346.12	74.31	686.07	242.32	230.97
	0.6	18.35	2627.73	115.93	352.05	2327.65	84.12	694.56	243.23	223.24
	0.8	2.76	2472.49	137.32	446.03	2601.77	74.75	655.86	248.85	235.58

TABLE VI

RUNNING TIME (IN SECONDS) OF TENSOR DATA CLUSTERING ON COIL20 AND UCSD DATASETS WITH DIFFERENT SRS AND NRS

Dataset	SR	LRSC	LMSC	RSS	RKLRR	IBDLR	S3COMP	TLRR	BDR	GNLTA	TTNN-DCT	TTNN-Data
COIL20 (NR = 0.1)	0.3	0.61	434.17	46.69	7862.62	3318.00	58.68	74.68	24.99	737.36	55.24	54.98
	0.5	1.15	434.06	44.70	3846.50	3747.00	57.99	72.91	23.70	1489.00	56.31	54.98
	0.7	1.11	429.08	44.02	276.79	3540.10	57.73	71.75	20.99	1508.20	56.80	55.91
UCSD (NR = 0.3)	0.3	0.48	249.82	26.40	6736.20	3198.00	44.46	66.01	10.81	884.03	47.05	24.35
	0.5	1.04	243.33	25.21	4827.70	3081.70	41.20	62.54	11.11	881.81	45.73	45.64
	0.7	1.24	243.87	27.46	6428.40	3361.50	41.94	62.25	9.73	887.43	45.86	45.56

We also report the running time of different methods in Tables V and VI. One can see that our methods both TTNN-DCT and TTNN-Data achieve the competitive performance in terms of time cost. Meanwhile, our methods take less time to achieve the best results in most cases.

VIII. CONCLUSION

In this work, we propose a novel transform tensor low-rank representation (TTLRR) method for tensor data recovery and subspace clustering with a theoretical guarantee. TTLRR can preserve the tensor structure information and better exploit the intrinsic low-rank representation of underlying data. More importantly, we prove the proposed method TTLRR can exactly recover the underlying clean data with a high probability; moreover, we provide the theoretical guarantee of the proposed model to ensure the clustering block-diagonal structure of the optimal solution. An effective ADMM algorithm is developed to solve the proposed model. Extensive numerical results on tensor data recovery and data clustering have demonstrated the effectiveness and robustness of the proposed method with DCT and data-dependent transforms. In the future, the nonlinear transform will be ulteriorly considered for the data recovery and clustering with recoverability study. Other learning tasks including supervised and semisupervised learning are also within the scope of the next research. Moreover, we will attempt to recover and cluster the corrupted multi-way visual data with large areas missing across all bands—that is a challenging task for our method.

REFERENCES

[1] C. Lu, H. Min, Z. Zhao, L. Zhu, D. Huang, and S. Yan, "Robust and efficient subspace segmentation via least squares regression," in *Proc. ECCV*, 2012, pp. 347–360.

[2] M. Tapaswi, M. Law, and S. Fidler, "Video face clustering with unknown number of clusters," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5027–5036.

[3] J. Wu et al., "Deep comprehensive correlation mining for image clustering," in *Proc. ICCV*, 2019, pp. 8149–8158.

[4] M. Ding, T.-Z. Huang, X.-L. Zhao, M. K. Ng, and T.-H. Ma, "Tensor train rank minimization with nonlocal self-similarity for tensor completion," *Inverse Problem Imag.*, vol. 15, no. 3, pp. 475–498, 2021.

[5] M. Yamaguchi, G. Irie, T. Kawanishi, and K. Kashino, "Subspace structure-aware spectral clustering for robust subspace clustering," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9874–9883.

[6] Q. Ma, S. Li, W. Zhuang, S. Li, J. Wang, and D. Zeng, "Self-supervised time series clustering with model-based dynamics," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 9, pp. 3942–3955, Sep. 2021.

[7] J.-H. Yang, X.-L. Zhao, T.-H. Ma, M. Ding, and T.-Z. Huang, "Tensor train rank minimization with hybrid smoothness regularization for visual data recovery," *Appl. Math. Model.*, vol. 81, pp. 711–726, May 2020.

[8] X. Chen, J. Z. Haung, F. Nie, R. Chen, and Q. Wu, "A self-balanced min-cut algorithm for image clustering," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2080–2088.

[9] Y. Chen, X. Xiao, Z. Hua, and Y. Zhou, "Adaptive transition probability matrix learning for multiview spectral clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 9, pp. 4712–4726, Sep. 2021.

[10] T. Wu and W. U. Bajwa, "A low tensor-rank representation approach for clustering of imaging data," *IEEE Signal Process. Lett.*, vol. 25, no. 8, pp. 1196–1200, Aug. 2018.

[11] Z. T. Chen, C. Chen, Z. B. Zheng, and Y. Zhu, "Tensor decomposition for multilayer networks clustering," in *Proc. AAAI*, 2019, vol. 33, no. 1, pp. 3371–3378.

[12] R. V. Lopes, J. Y. Ishihara, and G. A. Borges, "Identification of state-space switched linear systems using clustering and hybrid filtering," *J. Brazilian Soc. Mech. Sci. Eng.*, vol. 39, no. 2, pp. 565–573, Feb. 2017.

[13] R. Vidal and R. Hartley, "Motion segmentation with missing data using powerfactorization and GPCA," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2004, p. 2.

[14] C. Chen, Z.-B. Wu, Z.-T. Chen, Z.-B. Zheng, and X.-J. Zhang, "Auto-weighted robust low-rank tensor completion via tensor-train," *Inf. Sci.*, vol. 567, pp. 100–115, Aug. 2021.

[15] S. Zhang, X. Li, M. Zong, X. Zhu, and R. Wang, "Efficient kNN classification with different numbers of nearest neighbors," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 5, pp. 1774–1785, May 2018.

[16] M. Ding, X. Fu, T. Z. Huang, J. Wang, and X. L. Zhao, "Hyperspectral super-resolution via interpretable block-term tensor modeling," *IEEE J. Sel. Topics Signal Process.*, vol. 15, no. 3, pp. 641–656, Apr. 2021.

[17] Q. Xie, Q. Zhao, D. Meng, and Z. Xu, "Kronecker-basis-representation based tensor sparsity and its applications to tensor recovery," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 8, pp. 1888–1902, Aug. 2018.

[18] E. J. Candès and Y. Plan, "Matrix completion with noise," *Proc. IEEE*, vol. 98, no. 6, pp. 925–936, Jun. 2010.

[19] J. Fan and T. W. S. Chow, "Sparse subspace clustering for data with missing entries and high-rank matrix completion," *Neural Netw.*, vol. 93, pp. 36–44, Sep. 2017.

[20] Q. Shi, Y.-M. Cheung, Q. Zhao, and H. Lu, "Feature extraction for incomplete data via low-rank tensor decomposition with feature regularization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 6, pp. 1803–1817, Jun. 2019.

[21] X.-L. Zhao, J.-H. Yang, T.-H. Ma, T.-X. Jiang, M. K. Ng, and T.-Z. Huang, "Tensor completion via complementary global, local, and nonlocal priors," *IEEE Trans. Image Process.*, vol. 31, pp. 984–999, 2022.

[22] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 1, pp. 1–37, 2009.

[23] G. Liu, Q. Liu, and P. Li, "Blessing of dimensionality: Recovering mixture data via dictionary pursuit," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 1, pp. 47–60, Jan. 2017.

[24] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2765–2781, Mar. 2013.

[25] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171–184, Jan. 2013.

[26] K. Tang, R. Liu, Z. Su, and J. Zhang, "Structure-constrained low-rank representation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 12, pp. 2167–2179, Dec. 2014.

[27] Y. Caixia et al., "Self-weighted robust LDA for multiclass classification with edge classes," *ACM Trans. Intell. Syst. Technol.*, vol. 12, no. 1, pp. 1–19, 2020.

[28] B. Ran, H. Tan, J. Feng, W. Wang, Y. Cheng, and P. Jin, "Estimating missing traffic volume using low multilinear rank tensor completion," *J. Intell. Transp. Syst.*, vol. 20, no. 2, pp. 152–161, 2015.

[29] P. Giancarlo, "A low-rank tensor model for imputation of missing vehicular traffic volume," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8934–8938, Sep. 2018.

[30] J. Zhang et al., "Low-rank regularized heterogeneous tensor decomposition for subspace clustering," *IEEE Signal Process. Lett.*, vol. 25, no. 3, pp. 333–337, Mar. 2018.

[31] A. Johnson, J. Francis, B. Madathil, and S. N. George, "A two-way optimization framework for clustering of images using weighted tensor nuclear norm approximation," in *Proc. Nat. Conf. Commun. (NCC)*, Feb. 2020, pp. 1–5.

[32] X. J. Chang, F. P. Nie, S. Wang, Y. Yang, X. F. Zhou, and C. Q. Zhang, "Compound rank-k projections for bilinear analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 7, pp. 1502–1513, Jul. 2016.

[33] P. Zhou, C. Lu, J. Feng, Z. Lin, and S. Yan, "Tensor low-rank representation for data recovery and clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 5, pp. 1718–1732, May 2021.

[34] X. Zhang, D. Wang, Z. Zhou, and Y. Ma, "Robust low-rank tensor recovery with rectification and alignment," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 238–255, Jan. 2021.

[35] Y. Chen, S. Wang, C. Peng, Z. Hua, and Y. Zhou, "Generalized nonconvex low-rank tensor approximation for multi-view subspace clustering," *IEEE Trans. Image Process.*, vol. 30, pp. 4022–4035, 2021.

[36] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Proc. ECCV*, 2018, pp. 132–149.

[37] X. Ji, A. Vedaldi, and J. Henriques, "Invariant information clustering for unsupervised image classification and segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9865–9874.

[38] Y. Li, P. Hu, Z. Liu, D. Peng, J. T. Zhou, and X. Peng, "Contrastive clustering," in *Proc. AAAI Conf. Artif. Intell.*, May 2021, vol. 35, no. 10, pp. 8547–8555.

[39] X. Peng, J. Feng, S. Xiao, W.-Y. Yau, J. T. Zhou, and S. Yang, "Structured autoencoders for subspace clustering," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 5076–5086, Oct. 2018.

[40] X. Peng, J. Feng, J. T. Zhou, Y. Lei, and S. Yan, "Deep subspace clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 12, pp. 5509–5521, Dec. 2020.

[41] F. Tian, B. Gao, Q. Cui, E. Chen, and T. Y. Liu, "Learning deep representations for graph clustering," in *Proc. AAAI*, 2014, vol. 28, no. 1, pp. 1293–1299.

[42] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *Proc. Int. Conf. Mach. Learn.*, vol. 48, 2015, pp. 478–487.

[43] J. Yang, D. Parikh, and D. Batra, "Joint unsupervised learning of deep representations and image clusters," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5147–5156.

[44] Y. M. Asano, C. Rupprecht, and A. Vedaldi, "Self-labelling via simultaneous clustering and representation learning," in *Proc. ICLR*, 2020, pp. 1–22.

[45] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 208–220, Jan. 2013.

[46] L. R. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, no. 3, pp. 279–311, Sep. 1996.

[47] Z. Zhang and S. Aeron, "Exact tensor completion using t-SVD," *IEEE Trans. Signal Process.*, vol. 65, no. 6, pp. 1511–1526, Mar. 2016.

[48] Z. Huang, G. Zhou, and Y. Qiu, "Bayesian robust tensor ring model for incomplete multiway data," 2022, *arXiv:2202.13321*.

[49] Y.-Y. Liu, X.-L. Zhao, G.-J. Song, Y.-B. Zheng, and T.-Z. Huang, "Fully-connected tensor network decomposition for robust tensor completion problem," 2021, *arXiv:2110.08754*.

[50] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan, "Tensor robust principal component analysis with a new tensor nuclear norm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 4, pp. 925–938, Jan. 2020.

[51] J. Francis and S. N. George, "A unified tensor framework for clustering and simultaneous reconstruction of incomplete imaging data," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 16, no. 3, pp. 1–24, Sep. 2020.

[52] Y. Chen, A. Jalali, S. Sanghavi, and C. Caramanis, "Low-rank matrix recovery from errors and erasures," *IEEE Trans. Inf. Theory*, vol. 59, no. 7, pp. 4324–4337, Jul. 2013.

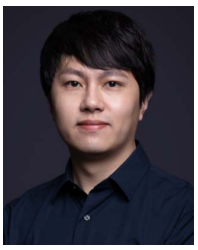
[53] J. L. Schafer and J. W. Graham, "Missing data: Our view of the state of the art," *Psychol. Methods*, vol. 7, no. 2, pp. 147–177, 2002.

[54] G. Song, M. K. Ng, and X. Zhang, "Robust tensor completion using transformed tensor singular value decomposition," *Numer. Linear Algebra Appl.*, vol. 27, no. 3, pp. 1–27, May 2020.

[55] M. Brbić and I. Kopriva, "Multi-view low-rank sparse subspace clustering," *Pattern Recognit.*, vol. 73, pp. 247–258, Jan. 2018.
[56] C. Lu, X. Peng, and Y. Wei, "Low-rank tensor completion with a new tensor nuclear norm induced by invertible linear transforms," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5996–6004.
[57] Q. Jiang and M. Ng, "Robust low-tubal-rank tensor completion via convex optimization," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 2649–2655.
[58] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.
[59] J. Eckstein and W. Yao, "Understanding the convergence of the alternating direction method of multipliers: Theoretical and computational perspectives," *Pacific J. Optim.*, vol. 11, no. 4, pp. 619–644, Jun. 2015.
[60] T. Zhang and Z. Shen, "A fundamental proof of convergence of alternating direction method of multipliers for weakly convex optimization," *J. Inequal. Appl.*, vol. 2019, no. 1, pp. 1–21, Dec. 2019.
[61] M. Tao and X. Yuan, "Convergence analysis of the direct extension of ADMM for multiple-block separable convex minimization," *Adv. Comput. Math.*, vol. 44, no. 3, pp. 773–813, Jun. 2018.
[62] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[63] K.-C. Lee, J. Ho, and D. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 684–698, May 2005.
[64] R. Vidal and P. Favaro, "Low rank subspace clustering (LRSC)," *Pattern Recognit. Lett.*, vol. 43, pp. 47–61, Jul. 2013.
[65] C. Zhang, Q. Hu, H. Fu, P. Zhu, and X. Cao, "Latent multi-view subspace clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4333–4341.
[66] X. Guo, "Robust subspace segmentation by simultaneously learning data representations and their affinity matrix," in *Proc. IJCAI*, 2015, pp. 3547–3553.
[67] S. Xiao, M. Tan, D. Xu, and Z. Y. Dong, "Robust kernel low-rank representation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 11, pp. 2268–2281, Nov. 2016.
[68] X. Xie, X. Guo, G. Liu, and J. Wang, "Implicit block diagonal low-rank representation," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 477–489, Jan. 2018.
[69] Y. Chen, C.-G. Li, and C. You, "Stochastic sparse subspace clustering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4155–4164.
[70] C. Lu, J. Feng, Z. Lin, T. Mei, and S. Yan, "Subspace clustering by block diagonal representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 2, pp. 487–501, Feb. 2019.
[71] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance," *J. Mach. Learn. Res.*, vol. 11, pp. 2837–2854, Jan. 2010.
[72] C. D. Manning, P. Raghavan, and H. Schutze, *Introduction to Information Retrieval*. Cambridge, U.K.: Cambridge Univ. Press, 2008.

Jing-Hua Yang (Graduate Student Member, IEEE) received the B.S. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2016. She is currently pursuing the Ph.D. degree with the Faculty of Information Technology, Macau University of Science and Technology, Macau.

Her current research interests include data mining, image processing, and artificial intelligence.



Chuan Chen received the Ph.D. degree from Hong Kong Baptist University, Hong Kong, in 2016.

He was a Post-Doctoral Researcher with the Department of Electrical Engineering, Katholieke Universiteit Leuven, Leuven, Belgium. He is currently an Associate Professor with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China. He has authored or coauthored over 50 international journals and conference papers. His research interests include numerical linear algebra, optimization, and their applications in machine learning.

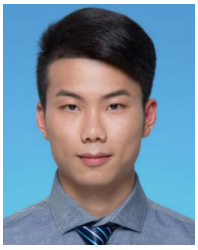
Hong-Ning Dai (Senior Member, IEEE) received the Ph.D. degree in computer science and engineering from the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong, in 2010.

He is currently an Associate Professor with the Department of Computer Science, Hong Kong Baptist University, Hong Kong. His current research interests include big data analytics, the Internet of Things, and blockchain technology.

Dr. Dai is a Senior Member of the Association for Computing Machinery. He has served as an Associate Editor/Editor for the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, *Ad Hoc Networks*, and *Connection Science*.

Meng Ding received the Ph.D. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2021.

From 2019 to 2020, he was an exchange Ph.D. Student at Oregon State University, Corvallis, OR, USA, supported by the China Scholarship Council. He is currently an Assistant Professor with the School of Mathematics, Southwest Jiaotong University, Chengdu. His current research interests include image processing, tensor analysis, and machine learning.



Zhe-Bin Wu received the B.S. degree majoring in information and computing science from Sun Yat-sen University, Guangzhou, China, in 2020, where he is currently pursuing the master's degree.

His current research interests include numerical optimization, numerical linear algebra, and machine learning.

Zibin Zheng (Senior Member, IEEE) received the Ph.D. degree from The Chinese University of Hong Kong, Hong Kong, in 2011.

He is currently a Professor with the School of Software Engineering, Sun Yat-sen University, China, where he also serves as the Chairperson of the Department of Software Engineering. He has authored or coauthored over 120 international journals and conference papers, including three Essential Science Indicators (ESI) highly cited papers. His research interests include blockchain, services computing, software engineering, and financial big data.

Dr. Zheng was a recipient of several awards, including the Top 50 Influential Papers in the Blockchain of 2018, the ACM SIGSOFT Distinguished Paper Award at ACM/IEEE International Conference on Software Engineering (ICSE2010), and the Best Student Paper Award at International Conference on Web Services (ICWS) 2010. He has served as the General Co-Chair for the BlockSys'19 and the CollaborateCom'16 and the PC Co-Chair for the 19th International Symposium on Cloud and Service Computing (SC2'19), the 18th International Congress on Internet of Things (ICIOT'18), and the 14th International Conference on Internet of Vehicles (IoV'14). According to Google Scholar, his papers have more than 7000 citations, with an H-index of 42.