Dictionary Learning With Low-Rank Coding Coefficients for Tensor Completion

Tai-Xiang Jiang, Xi-Le Zhao, Hao Zhang, and Michael K. Ng

***Abstract*— In this article, we propose a novel tensor learning and coding model for third-order data completion. The aim of our model is to learn a data-adaptive dictionary from given observations and determine the coding coefficients of third-order tensor tubes. In the completion process, we minimize the low-rankness of each tensor slice containing the coding coefficients. By comparison with the traditional predefined transform basis, the advantages of the proposed model are that: 1) the dictionary can be learned based on the given data observations so that the basis can be more adaptively and accurately constructed and 2) the low-rankness of the coding coefficients can allow the linear combination of dictionary features more effectively. Also we develop a multiblock proximal alternating minimization algorithm for solving such tensor learning and coding model and show that the sequence generated by the algorithm can globally converge to a critical point. Extensive experimental results for real datasets such as videos, hyperspectral images, and traffic data are reported to demonstrate these advantages and show that the performance of the proposed tensor learning and coding method is significantly better than the other tensor completion methods in terms of several evaluation metrics.**

***Index Terms*— Dictionary learning, low-rank coding, tensor completion, tensor singular value decomposition (t-SVD).**

I. Introduction

TENSOR completion is a problem of filling the missing or unobserved entries of incomplete observed data, playing an important role in a wide range of real-world applications, such as color image inpainting [1]–[5], high-speed compressive video [6], magnetic resonance imaging (MRI) data recovery [7], and hyperspectral data inpainting [8]. Generally, many real-world tensors are inner correlated, for example, the spectral redundancy [9]–[11] of the hyperspectral images (HSIs). Therefore, it is effective to use the global low-dimensional structure to characterize the relationship between the missing entries and the observed ones.

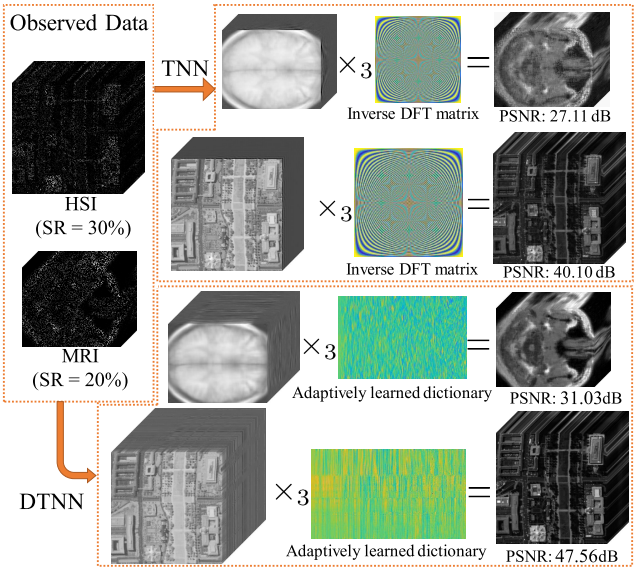


Fig. 1. Illustration of the TNN-based LRTC and the DTNN-based LRTC. SR denotes the sampling rate.

Generally, like the matrix case, the low-rank tensor completion (LRTC) can be formulated as

$$\min \ \text{rank}(\mathcal{X}) \quad \text{s.t. } \mathcal{X}_\Omega = \mathcal{O}_\Omega \tag{1}$$

where $\mathcal{X}$ is the underlying tensor, $\mathcal{O}$ is the observed incomplete tensor as shown in the top left of Fig. 1, Ω is the index set corresponding to the observed entries, and $\mathcal{X}_\Omega = \mathcal{O}_\Omega$ enforces the entries of $\mathcal{X}$ in Ω equal to the observation $\mathcal{O}$. However, unlike the matrix cases, the definition of the tensor rank is still not unique and has received considerable attentions in recent researches. Generally, different definitions of the tensor rank are respectively based on different tensor decomposition schemes. For instance, the CANDECOMP/PARAFAC (CP) rank, based on the CP decomposition, is defined as the minimal rank-one tensors to express the original data [12]. Although determinating the CP rank of a given tensor is NP-hard [13], CP decomposition has been successfully applied for tensor

Manuscript received 4 September 2020; revised 27 February 2021 and 1 June 2021; accepted 11 August 2021. Date of publication 31 August 2021; date of current version 6 February 2023. This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant JBK2102001; in part by the National Natural Science Foundation of China under Grant 12001446, Grant 61772003, and Grant 61876203; in part by the Applied Basic Research Project of Sichuan Province under Grant 2021YJ0107; in part by the Key Project of Applied Basic Research in Sichuan Province under Grant 2020YJ0216; in part by the National Key Research and Development Program of China under Grant 2020YFA0714001; in part by HKRGC GRF under Grant 12200317, Grant 12300218, Grant 12300519, and Grant 17201020; and in part by the Financial Intelligence and Financial Engineering Research Key Laboratory of Sichuan province. *(Corresponding author: Michael K. Ng.)*

Tai-Xiang Jiang is with the FinTech Innovation Center, School of Economic Information Engineering, Southwestern University of Finance and Economics, Chengdu, Sichuan 611130, China (e-mail: taixiangjiang@gmail.com; jiangtx@swufe.edu.cn).

Xi-Le Zhao and Hao Zhang are with the Research Center for Image and Vision Computing, School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: xlzhao122003@163.com; 201821110218@std.uestc.edu.cn).

Michael K. Ng is with the Department of Mathematics, The University of Hong Kong, Pokfulam, Hong Kong (e-mail: mng@maths.hku.hk).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2021.3104837.

Digital Object Identifier 10.1109/TNNLS.2021.3104837

recovery problem [14]–[16]. The Tucker rank, corresponding to the Tucker decomposition [17], is defined as a vector constituted of the ranks of the unfolding matrices along all modes. Liu *et al.* [4] propose a convex surrogate of the Tucker rank and minimize it for the LRTC problem, while Zhang [18] resort to using a family of nonconvex functions onto the singular values. Another newly emerged one is the tensor train (TT) rank derived from TT decomposition [19]. In this framework, the tensor is decomposed in a chain manner with nodes being third-order tensors. Bengua *et al.* [20] minimize a nuclear norm based on the TT rank for color image and video recovery. TT rank has also been applied for HSI super-resolution [21] and tensor-on-tensor regression [22]. When factors are cyclically connected, they become tensor ring (TR) decomposition [23]. Yuan *et al.* [24] exploit the low-rank structure of the TR latent space and regularize the latent TR factors with the nuclear norm. Yu *et al.* [25] introduce tensor circular unfolding for TR decomposition and perform parallel low-rank matrix factorizations to all circularly unfolded matrices for tensor completion. Please refer to [26] and [27] for a comprehensive overview of the LRTC problem.

This work fixes attentions on novel notions of the tensor rank, i.e., the tensor tubal rank and multirank, which are derived from the tensor singular value decomposition (t-SVD) framework [28]–[30]. The t-SVD framework is constructed based on a fundamental tensor-tensor product (t-prod) operation (see Definition 2), which is closed on the set of third-order tensors and allows tensor factorizations which are analogs of matrix factorizations such as SVD. Meanwhile, it further allows new extensions of familiar matrix analysis to the multilinear setting while avoiding the loss of information inherent in matricization or flattening of the third-order tensor [31]. For a third-order tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, its t-SVD is given as $\mathcal{X} = \mathcal{U} * \mathcal{S} * \mathcal{V}^{\mathrm{H}}$, where $\mathcal{U} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$ and $\mathcal{V} \in \mathbb{R}^{n_2 \times n_2 \times n_3}$ are the orthogonal tensors, $\mathcal{S} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is an f-diagonal tensor (see Definition. 4), and $*$ denotes the t-prod (see Definition 2). The tensor tubal rank of $\mathcal{X}$ is defined as the number of nonzero singular tubes of $\mathcal{S}$. Because the LRTC problem associated with the tensor tubal rank (or multirank) is NP-hard, Zhang *et al.* [32] turn to minimize the tensor nuclear norm (TNN) (see Definition 8), which is a convex envelope of the ℓ_1 norm of the tensor multirank, and they establish theoretical guarantee in [33]. Jiang and Ng [34] and Wang *et al.* [35] tackle the robust tensor completion task, in which incomplete observations are corrupted by sparse outliers, via minimizing TNN. The TNN-based LRTC model is given as

$$\min \ \|\mathcal{X}\|_{\mathrm{TNN}} \quad \text{s.t. } \mathcal{X}_\Omega = \mathcal{O}_\Omega. \tag{2}$$

As the t-prod is based on a convolution-like operation, the computation of t-prod and TNN could be implemented with discrete Fourier transform (DFT) or fast Fourier transform (FFT).

Kernfeld *et al.* [36] further note that a more general t-product could be defined with any invertible linear transforms. Also, the TNN in (2) can be alternatively constructed using other transform, e.g., the discrete cosine transform (DCT) adopted by Lu *et al.* [37] and Xu *et al.* [38], and the Haar wavelet transform exploited in [39]. Furthermore, Jiang *et al.* [40] introduce the framelet transform, which is semi-invertible, and break through the restriction of invertibility. Within these transform-based TNN methods, the typical pipeline is applying one selected transform along the third dimension and minimizing the low-rankness of slices of the transformed data for completion. Once the tubes of the original tensor are highly correlated, the frontal slices of the transformed data would be low-rank [39], [40].

An unavoidable issue is that the correlations along the third mode are different for various types of data. For example, the redundancy of HSIs along the third mode is much higher than videos with changing scenes. Thus, predefined transforms usually lack flexibility and are not suitable for all kinds of data. Therefore, to address this issue, we construct a dictionary, which can be adaptively inferred form the data, instead of inverse transforms mentioned above. As mentioned by Lu *et al.* and Song *et al.*, it is interesting to learn the transform for implementing t-SVD from the data in different tasks. Our approach can be viewed as learning the inverse transform from this perspective and indeed enriches the research on this topic. The methods, which use DFT, DCT, and Framelet, can be viewed as specific instances of our method with fixed dictionaries, i.e., the inverse discrete transformation matrices. From the view of dictionary learning, our method can also be interpreted as learning a dictionary with low-rank coding. We enforce the low-rankness of the coding coefficients in a tensor manner, and this allows the linear combination of features, namely, the atoms of the dictionary.

The main contributions of this article mainly consist of three aspects.

1) We propose novel tensor learning and coding model, which is to adaptively learn a dictionary from the observations and determine the low-rank coding coefficients, for the third-order tensor completion.
2) A multiblock proximal alternating minimization algorithm is designed to solve the proposed nonconvex model. We theoretically prove its global convergence to a critical point.
3) Extensive experiments are conducted on various types of real-world third-order tensor data. The results illustrate that our method outperforms compared with LRTC methods.

This article is organized as follows. Section II introduces related works and basic preliminaries. Our method is given in Section III. We report the experimental results in Section IV. Finally, Section V draws some conclusions.

II. Preliminaries

Throughout this article, lowercase letters, e.g., x, boldface lowercase letters, e.g., $\mathbf{x}$, boldface upper case letters, e.g., $\mathbf{X}$, and boldface calligraphic letters, e.g., $\mathcal{X}$, are used to denote scalars, vectors, matrices, and tensors, respectively. Given a third-order tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, we use $\mathcal{X}_{ijk}$ to denote its (i, j, k)th element. The kth frontal slice of $\mathcal{X}$ is denoted as $\mathcal{X}^{(k)}$ (or $\mathcal{X}(:, :, k)$, $\mathbf{X}^k$), and the mode-3 unfolding matrix of $\mathcal{X}$ is denoted as $\mathbf{X}_{(3)} \in \mathbb{R}^{n_3 \times n_1 n_2}$. We use $\mathtt{fold}_3$ and $\mathtt{unfold}_3$ to denote the folding and unfolding operations along the third dimension, respectively, and we have

$\mathcal{X} = \texttt{fold}_3(\texttt{unfold}_3(\mathcal{X})) = \texttt{fold}_3(\mathbf{X}_{(3)})$. The mode-3 tensor-matrix product is denoted as $\times_3$, and we have $\mathcal{X} \times_3 \mathbf{A} \Leftrightarrow \mathbf{A}\texttt{unfold}_3(\mathcal{X})$. The DFT matrix and inverse DFT matrix for a vector of the length n are, respectively, denoted as $\mathbf{F}_n$ and $\mathbf{F}_n^{-1}$. For the tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, its Fourier-transformed (along the third mode) tensor $\mathcal{Z} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$ can be obtained by $\mathcal{Z} = \mathcal{X} \times_3 \mathbf{F}_{n_3}$, and we have $\mathcal{X} = \mathcal{Z} \times_3 \mathbf{F}_{n_3}^{-1}$. The tensor Frobenius norm of a third-order tensor $\mathcal{X}$ is defined as $\|\mathcal{X}\|_F := (\langle \mathcal{X}, \mathcal{X} \rangle)^{1/2} = (\sum_{ijk} \mathcal{X}_{ijk}^2)^{1/2}$. For a matrix $\mathbf{X} \in \mathbb{C}^{n_1 \times n_2}$, its matrix nuclear norm is denoted as $\|\mathbf{X}\|_* = \sum_{i=1}^{\min\{n_1, n_2\}} \sigma_i(\mathbf{X})$, where $\sigma_i(\mathbf{X})$ is the ith largest singular value of $\mathbf{X}$.

Definition 1 (Tensor Conjugate Transpose [31]): The conjugate transpose of a tensor $\mathcal{A} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$ is tensor $\mathcal{A}^{\mathrm{H}} \in \mathbb{C}^{n_2 \times n_1 \times n_3}$ obtained by conjugate transposing each of the frontal slice and then reversing the order of transposed frontal slices 2 through n_3, i.e., $(\mathcal{A}^{\mathrm{H}})^{(1)} = (\mathcal{A}^{(1)})^{\mathrm{H}}$ and $(\mathcal{A}^{\mathrm{H}})^{(i)} = (\mathcal{A}^{(n_3+2-i)})^{\mathrm{H}}$ for $i = 2, \ldots, n_3$.

Definition 2 (t-Prod [31]): The t-prod $\mathcal{C} = \mathcal{A} * \mathcal{B}$ of $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{B} \in \mathbb{R}^{n_2 \times n_4 \times n_3}$ is a tensor of size $n_1 \times n_4 \times n_3$, where the (i, j)th tube $\mathbf{c}_{ij:}$ is given by

$$\mathbf{c}_{ij:} = \mathcal{C}(i, j, :) = \sum_{k=1}^{n_2} \mathcal{A}(i, k, :) \circledast \mathcal{B}(k, j, :) \tag{3}$$

where $\circledast$ denotes the circular convolution between two tubes of same size.

Equivalently, for $\mathcal{C} = \mathcal{A} * \mathcal{B}$, we have

$$\begin{pmatrix} \mathcal{C}^{(1)} \\ \mathcal{C}^{(2)} \\ \vdots \\ \mathcal{C}^{(n_3)} \end{pmatrix} = \begin{pmatrix} \mathcal{A}^{(1)} & \mathcal{A}^{(n_3)} & \cdots & \mathcal{A}^{(2)} \\ \mathcal{A}^{(2)} & \mathcal{A}^{(1)} & \cdots & \mathcal{A}^{(3)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{A}^{(n_3)} & \mathcal{A}^{(n_3-1)} & \cdots & \mathcal{A}^{(1)} \end{pmatrix} \begin{pmatrix} \mathcal{B}^{(1)} \\ \mathcal{B}^{(2)} \\ \vdots \\ \mathcal{B}^{(n_3)} \end{pmatrix}$$

where the first item in the right part of the equation is also called the block circulant unfolding of $\mathcal{A}$.

Definition 3 (Facewise Product [36]): For two third-order tensors $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{B} \in \mathbb{R}^{n_2 \times n_4 \times n_3}$, their facewise product $\mathcal{A} \triangle \mathcal{B} \in \mathbb{R}^{n_1 \times n_4 \times n_3}$ is defined according to

$$(\mathcal{A} \triangle \mathcal{B})^{(i)} = \mathcal{A}^{(i)} \mathcal{B}^{(i)}, \quad \text{for } i = 1, 2, \ldots, n_3.$$

As the convolution operation could be converted into elementwise product via Fourier transform, we have

$$\mathcal{C} = \mathcal{A} * \mathcal{B} = \left(\left(\mathcal{A} \times_3 \mathbf{F}_{n_3}\right) \triangle \left(\mathcal{B} \times_3 \mathbf{F}_{n_3}\right)\right) \times_3 \mathbf{F}_{n_3}^{-1}. \tag{4}$$

Equation (4) indicates that we can compute t-prod between two tensors using the DFT matrix or FFT for acceleration.

Definition 4 (Special Tensors [31]): The **identity** tensor $\mathcal{I} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$ is the tensor whose first frontal slice is the $n_1 \times n_1$ identity matrix, and whose other frontal slices are all zeros. A tensor $\mathcal{Q} \in \mathbb{C}^{n_1 \times n_1 \times n_3}$ is **orthogonal** if it satisfies

$$\mathcal{Q}^{\mathrm{H}} * \mathcal{Q} = \mathcal{Q} * \mathcal{Q}^{\mathrm{H}} = \mathcal{I}. \tag{5}$$

A tensor $\mathcal{A}$ is called **f-diagonal** if each frontal slice $\mathcal{A}^{(i)}$ is a diagonal matrix.

Theorem 1 (t-SVD [29], [31]): For $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the t-SVD of $\mathcal{A}$ is given by

$$\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^{\mathrm{H}} \tag{6}$$

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

Definition 5 (Tensor Tubal Rank [32]): The tensor tubal rank of a tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, denoted as $\text{rank}_t(\mathcal{A})$, is defined as the number of nonzero singular tubes in $\mathcal{S}$, where $\mathcal{S}$ is from the t-SVD of $\mathcal{A}$: $\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^{\mathrm{H}}$. Formally, we can write

$$\text{rank}_t(\mathcal{A}) = \#\{i, \mathcal{S}(i, i, :) \neq 0\}.$$

An alternative definition of the tensor tubal rank is that it is the largest rank of all the frontal slices of $\mathcal{A} \times_3 \mathbf{F}_{n_3}$ in Fourier domain.

Suppose the tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ has tensor tubal rank r, then the reduced t-SVD of $\mathcal{A}$ is given by $\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^{\mathrm{H}}$, where $\mathcal{U} \in \mathbb{R}^{n_1 \times r \times n_3}$ and $\mathcal{V} \in \mathbb{R}^{r \times n_2 \times n_3}$ are the orthogonal tensors, and $\mathcal{S} \in \mathbb{R}^{r \times r \times n_3}$ is an f-diagonal tensor. An important property of t-SVD is that the truncated t-SVD of a tensor provides the optimal approximation measured by the Frobenius norm with the tubal rank at most r [31].

Definition 6 (Tensor Multirank [32]): Let $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ be a third-order tensor, the tensor multirank, denoted as $\text{rank}_{\mathrm{m}}(\mathcal{A}) \in \mathbb{R}^{n_3}$, is a vector whose ith element is the rank of the ith frontal slice of $\mathcal{B} = \mathcal{A} \times_3 \mathbf{F}_{n_3}$. We can write

$$\text{rank}_{\mathrm{m}}(\mathcal{A}) = \left[\text{rank}\left(\mathcal{B}^{(1)}\right), \text{rank}\left(\mathcal{B}^{(2)}\right), \ldots, \text{rank}\left(\mathcal{B}^{(n_3)}\right)\right]. \tag{7}$$

Given a third-order tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, we can find that its tensor tubal rank $\text{rank}_t(\mathcal{A})$ equals to the ℓ_∞ norm (or say the biggest value) of the tensor multirank $\text{rank}_m(\mathcal{A})$. As $\|\text{rank}_m(\mathcal{X})\|_1 \geq \|\text{rank}_m(\mathcal{X})\|_\infty = \text{rank}_t(\mathcal{X}) \geq (1/n_3)\|\text{rank}_m(\mathcal{X})\|_1$, the tensor tubal rank is bounded by the ℓ_1 norm of the tensor multirank.

Definition 7 (Block Diagonal Operation [32]): The block diagonal operation of $\mathcal{A} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$ is given by

$$\texttt{bdiag}(\mathcal{A}) \triangleq \begin{bmatrix} \mathcal{A}^{(1)} & & & \\ & \mathcal{A}^{(2)} & & \\ & & \ddots & \\ & & & \mathcal{A}^{(n_3)} \end{bmatrix} \tag{8}$$

where $\texttt{bdiag}(\mathcal{A}) \in \mathbb{C}^{n_1 n_3 \times n_2 n_3}$.

Definition 8 (TNN [32]): The TNN of a tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, denoted as $\|\mathcal{A}\|_{\text{TNN}}$, is defined as

$$\|\mathcal{A}\|_{\text{TNN}} \triangleq \left\| \begin{pmatrix} \mathcal{A}^{(1)} & \mathcal{A}^{(n_3)} & \cdots & \mathcal{A}^{(2)} \\ \mathcal{A}^{(2)} & \mathcal{A}^{(1)} & \cdots & \mathcal{A}^{(3)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{A}^{(n_3)} & \mathcal{A}^{(n_3-1)} & \cdots & \mathcal{A}^{(1)} \end{pmatrix} \right\|_*. \tag{9}$$

The TNN can be computed via the summation of the matrix nuclear norm of $(\mathcal{A} \times \mathbf{F}_{n_3})$'s frontal slices. That is, $\|\mathcal{A}\|_{\text{TNN}} = \sum_{i=1}^{n_3} \|(\mathcal{A} \times \mathbf{F}_{n_3})^{(i)}\|_*$.

Denoting the Fourier-transformed tensor of $\mathcal{X}$ as $\mathcal{Z} = \mathcal{X} \times \mathbf{F}_{n_3}$, we have $\mathcal{X} = \mathcal{Z} \times_3 \mathbf{F}_{n_3}^{-1}$ and the the equivalent form of the TNN-based LRTC model in (2) as

$$\min_{\mathcal{Z}} \ \|\texttt{bdiag}(\mathcal{Z})\|_* \ \text{ s.t. } \left(\mathcal{Z} \times_3 \mathbf{F}_{n_3}^{-1}\right)_\Omega = \mathcal{O}_\Omega. \tag{10}$$

III. Main Results

A. Proposed Model

As (10) can be interpreted as to find a slice-wisely low-rank coding of $\mathcal{X}$ with a predefined dictionary $\mathbf{F}_{n_3}^{-1}$, to promote flexibility, we replace $\mathbf{F}_{n_3}^{-1}$ with a data-adaptive dictionary and our tensor learning and coding model is formulated as

$$\begin{aligned} &\min_{\mathcal{Z},\mathbf{D}} \ \|\mathtt{bdiag}(\mathcal{Z})\|_* \\ &\text{s.t. } (\mathcal{Z}\times_3 \mathbf{D})_\Omega = \mathcal{O}_\Omega \\ &\qquad \|\mathbf{D}(:,i)\|_2 = 1 \quad \text{for } i = 1,2,\ldots,d, \end{aligned} \tag{11}$$

where $\mathbf{D} \in \mathbb{R}^{n_3\times d}$ and $\mathcal{Z} \in \mathbb{R}^{n_1\times n_2\times d}$ are the respective dictionary and low-rank coding coefficients. As our LRTC model is very similar to the TNN-based model in (10), we term it as dictionary-based TNN (DTNN).

On the one hand, if the dictionary $\mathbf{D} \in \mathbb{R}^{n_3\times d}$ is prefixed and there is a matrix $\mathbf{D}^* \in \mathbb{R}^{d\times n_3}$ which satisfies $\mathbf{D}\mathbf{D}^* = \mathbf{I}_{n_3}$, the structure of t-prod (and t-SVD) still holds when replacing $\mathbf{F}_{n_3}$ and $\mathbf{F}_{n_3}^{-1}$ in (4) with $\mathbf{D}^*$ and $\mathbf{D}$, respectively. The circular convolution operation between tubes, which is used to define the original t-prod, will change according to how $\mathbf{D}$ and $\mathbf{D}^*$ are constructed. Thus, a novel-type t-prod and t-SVD could be defined with better data flexibility and the performance would be promising. Meanwhile, if $d = n_3$ and $\mathbf{D}\mathbf{D}^* = \mathbf{D}\mathbf{D}^* = \mathbf{I}_{n_3}$, the exact recovery of the underlying tensor from random samplings is theoretical guaranteed under certain conditions [37].

On the other hand, although the objective function of (11) is in the same form of (10), it could not be derived to a normative definition of a norm as Definition 8 if finding $\mathbf{D}^*$ is difficult. Given a tensor $\mathcal{X}$ and a certain dictionary $\mathbf{D}$, the coefficients in $\mathcal{Z}$ here could not be directly obtained with satisfying $(\mathcal{Z}\times_3 \mathbf{D}) = \mathcal{X}$. It is needed to optimize (11) simultaneously with respect to the dictionary and coefficients (with Ω indexing all the entries). By the way, our DTNN can also be generalized for higher order tensors via the techniques proposed in [41] and [42], or for other applications, such as the tensor robust principal component analysis [43] and remote sensing images recovery [44], [45].

While resembling (10) in form, our model in (11) is distinct from the TNN-based LRTC model. The main difference is that our model is more flexible for different kinds of data because of the data-adaptive dictionary term. The bottom-right part in Fig. 1 shows the coefficients and the dictionaries obtained by our method. We can see that the dictionary learned for the HSI completion is smoother than that for the MRI data. With the adaptively learned dictionary and the corresponding low-rank coding, the performance of our method is significantly better than the TNN-based LRTC method. The dictionary used in (11) can be viewed as inverse transform. This is also different from previous works tailoring the linear or unitary transform [37], [39].

Traditional dictionary learning techniques use overcomplete dictionaries, the amount of whose atoms is always much more than the dimension of the signal, and find the sparse representations [46]. In (11), although d is much bigger than n_3, $\mathbf{D}$ is still not big enough to overcompletely represent $\mathcal{X}$, which is of a big volume, with sparse coefficients. Therefore, we need specific low-rank structure of the coefficients, which allows the linear combination of features, together with the learned dictionary, to accurately complete $\mathcal{X}$. Thus, our method is distinct from previous tensor dictionary learning methods, which enforce the sparsity or tubal sparsity of coefficients [47], [48]. Please see Section IV-E1 for detailed comparisons of sparsity and low-rankness.

B. Proposed Algorithm

To optimize the specific structured problem in the proposed model, we tailored a multiblock proximal alternating minimization algorithm. Let

$$\Phi(\mathcal{X}) = \begin{cases} 0, & \mathcal{X}_\Omega = \mathcal{O}_\Omega \\ \infty, & \text{otherwise} \end{cases}$$

and

$$\Psi(\mathbf{D}) = \begin{cases} 0, & \|\mathbf{D}(:,i)\|_2 = 1 \quad \text{for } i = 1,2,\ldots,d \\ \infty, & \text{otherwise.} \end{cases}$$

Thus, the problem in (11) can be rewritten as the following unconstraint problem:

$$\min_{\mathcal{Z},\mathbf{D}} \quad \Phi(\mathcal{Z}\times_3 \mathbf{D}) + \sum_{i=1}^{d} \left\|\mathcal{Z}^{(i)}\right\|_* + \Psi(\mathbf{D}). \tag{12}$$

As the minimization problem in (12) is difficult to be directly optimized, therefore, we resort to the half quadratic splitting (HQS) technique [49], [50] and turn to solve the following problem:

$$\min_{\mathcal{Z},\mathbf{D},\mathcal{X}} \frac{\beta}{2}\|\mathcal{X} - \mathcal{Z}\times_3 \mathbf{D}\|_F^2 + \Phi(\mathcal{X}) + \sum_{i=1}^{d} \left\|\mathcal{Z}^{(i)}\right\|_* + \Psi(\mathbf{D}). \tag{13}$$

We denote the objective function in (13) as $L(\mathcal{Z},\mathbf{D},\mathcal{X})$. The optimization problem in (13) is nonconvex and has more than two blocks. Thus, it prevents us from directly using some classical algorithms designed for convex optimizations, such as the alternating direction method of multipliers (ADMM) [51] used in [33], with theoretical convergence guarantees. We use the proximal alternating minimization framework [52] for this nonconvex problem with guaranteed convergence. In our algorithm, each variable is alternatively updated as

$$\begin{aligned} \mathcal{Z}_{k+1} &\in \arg\min_{\mathcal{Z}} \left\{ L(\mathcal{Z},\mathbf{D}_k,\mathcal{X}_k) + \frac{\rho_k^z}{2}\|\mathcal{Z}-\mathcal{Z}_k\|_F^2 \right\} \\ \mathbf{D}_{k+1} &\in \arg\min_{\mathbf{D}} \left\{ L(\mathcal{Z}_{k+1},\mathbf{D},\mathcal{X}_k) + \frac{\rho_k^d}{2}\|\mathbf{D}-\mathbf{D}_k\|_F^2 \right\} \\ \mathcal{X}_{k+1} &\in \arg\min_{\mathcal{X}} \left\{ L(\mathcal{Z}_{k+1},\mathbf{D}_{k+1},\mathcal{X}) + \frac{\rho_k^x}{2}\|\mathcal{X}-\mathcal{X}_k\|_F^2 \right\} \end{aligned} \tag{14}$$

where $(\rho_k^z)_{k\in\mathbb{N}}$, $(\rho_k^d)_{k\in\mathbb{N}}$, and $(\rho_k^x)_{k\in\mathbb{N}}$ are three positive sequences and $\mathcal{Z}_k$, $\mathbf{D}_k$, and $\mathcal{Z}_k$, respectively, denote the values of $\mathcal{Z}$, $\mathbf{D}$, and $\mathcal{Z}$ at the kth iteration. Thus, for example, $L(\mathcal{Z},\mathbf{D}_k,\mathcal{X}_k)$ is a function of $\mathcal{Z}$, which comes from

$L(\mathcal{Z}, \mathbf{D}, \mathcal{X})$ by fixing other two variables $\mathcal{X}$ and $\mathbf{D}$ as $\mathcal{X}_k$ and $\mathbf{D}_k$, respectively.

1) Updating $\mathcal{Z}$ and $\mathbf{D}$: Following the updating strategy in [53], the coefficient $\mathcal{Z}$ (or equivalently denoted as $\mathbf{Z}_{(3)}$ for simplification) and the dictionary $\mathbf{D}$ at the kth iteration can be, respectively, decomposed as follows:

$$\mathbf{Z}_{k(3)} = \begin{bmatrix} \mathbf{z}_k^{1\top} \\ \vdots \\ \mathbf{z}_k^{i\top} \\ \vdots \\ \mathbf{z}_k^{d\top} \end{bmatrix} = \begin{bmatrix} \mathtt{vec}\big(\mathbf{Z}_k^1\big)^\top \\ \vdots \\ \mathtt{vec}\big(\mathbf{Z}_k^i\big)^\top \\ \vdots \\ \mathtt{vec}\big(\mathbf{Z}_k^d\big)^\top \end{bmatrix} = \begin{bmatrix} \mathtt{vec}(\mathcal{Z}_k(:,:,1))^\top \\ \vdots \\ \mathtt{vec}(\mathcal{Z}_k(:,:,i))^\top \\ \vdots \\ \mathtt{vec}(\mathcal{Z}_k(:,:,d))^\top \end{bmatrix} \tag{15}$$

and

$$\mathbf{D}_k = \big[\mathbf{d}_k^1, \ldots, \mathbf{d}_k^i, \ldots, \mathbf{d}_k^d\big] \tag{16}$$

where $\mathbf{Z}_k^i = \mathcal{Z}_k(:,:,i)$ indicates the ith frontal slice of the coefficients' tensor $\mathcal{Z}$ at the kth iteration, $\mathbf{z}_k^i = \mathtt{vec}(\mathbf{Z}_k^i)$ $\mathtt{vec}(\cdot)$ denotes the vectorization operation, and $\mathbf{d}_k^i = \mathbf{D}_k(:,i)$ is the ith atom of $\mathbf{D}_k$. In our algorithm, the frontal slices of $\mathcal{Z}$ are frequently reshaped into vectors and vice versa. Therefore, we use $\mathtt{vec}(\cdot)$ to denote the vectorization from the frontal slices of $\mathcal{Z}$ to a column vector by stacking the columns of $\mathcal{Z}$, and $\mathtt{vec}(\cdot)^{-1}$ to denote inverse operation.

Thus, the $\mathcal{Z}$ subproblem and the $\mathbf{D}$ subproblem can be, respectively, split into d problems. Then, we update the pair of $\mathbf{Z}_{k+1}^i$ and $\mathbf{d}_{k+1}^i$ from $i = 1$ to d. This updating scheme is the same as the well-known KSVD technique [54]. From the decompositions in (15) and (16), at the beginning of the kth iteartion, we can rewrite the first term in the objective function as $(\beta/2)\|\mathcal{X}_k - \mathcal{Z}_k \times_3 \mathbf{D}_k\|_F^2 = (\beta/2)\|\mathbf{X}_{k(3)} - \mathbf{D}_k\mathbf{Z}_{k(3)}\|_F^2 = (\beta/2)\|\mathbf{X}_{k(3)} - \sum_{i=1}^d \mathbf{d}_k^i \mathbf{z}_k^{i\top}\|_F^2$. Thus, for simplicity, we introduce an intermediate variable as

$$\mathbf{R}_k^i = \mathbf{X}_{k(3)} - \sum_{j=1}^{i-1} \mathbf{d}_{k+1}^j \Big(\mathbf{z}_{k+1}^j\Big)^\top - \sum_{j=i+1}^{d} \mathbf{d}_k^j \Big(\mathbf{z}_k^j\Big)^\top. \tag{17}$$

Then, we solve following problems:

$$\mathbf{Z}_{k+1}^i = \arg\min_{\mathbf{Z}} \ \frac{\beta}{2}\big\|\mathbf{R}_k^i - \mathbf{d}_k^i \mathtt{vec}(\mathbf{Z})^\top\big\|_F^2 + \|\mathbf{Z}\|_* + \frac{\rho_k^z}{2}\big\|\mathbf{Z} - \mathbf{Z}_k^i\big\|_F^2 \tag{18}$$

and

$$\mathbf{d}_{k+1}^i = \arg\min_{\mathbf{d}} \ \frac{\beta}{2}\big\|\mathbf{R}_k^i - \mathbf{d}\big(\mathbf{z}_k^i\big)^\top\big\|_F^2 + \Psi(\mathbf{d}) + \frac{\rho_k^d}{2}\big\|\mathbf{d} - \mathbf{d}_k^i\big\|_F^2. \tag{19}$$

After denoting $\mathbf{z} = \mathtt{vec}(\mathbf{Z})$, two quadratic terms in (18) can be combined as

$$\begin{aligned}
&\frac{\beta}{2}\big\|\mathbf{R}_k^i - \mathbf{d}_k^i \mathbf{z}^\top\big\|_F^2 + \frac{\rho_k^z}{2}\big\|\mathbf{Z} - \mathbf{Z}_k^i\big\|_F^2 \\
&= \frac{\beta}{2}\big(\big\langle \mathbf{R}_k^i, \mathbf{R}_k^i\big\rangle - 2\big\langle \mathbf{R}_k^i, \mathbf{d}_k^i \mathbf{z}^\top\big\rangle + \big\langle \mathbf{d}_k^i \mathbf{z}^\top, \mathbf{d}_k^i \mathbf{z}^\top\big\rangle\big) \\
&\quad + \frac{\rho_k^z}{2}\big(\langle \mathbf{Z}, \mathbf{Z}\rangle - 2\big\langle \mathbf{Z}, \mathbf{Z}_k^i\big\rangle + \big\langle \mathbf{Z}_k^i, \mathbf{Z}_k^i\big\rangle\big) \\
&= \frac{\beta}{2}\Big(\big\langle \mathbf{R}_k^i, \mathbf{R}_k^i\big\rangle - 2\Big\langle \mathtt{vec}^{-1}\Big(\big(\mathbf{R}_k^i\big)^\top \mathbf{d}_k^i\Big), \mathbf{z}\Big\rangle + \langle \mathbf{z}, \mathbf{z}\rangle\Big) \\
&\quad + \frac{\rho_k^z}{2}\big(\langle \mathbf{Z}, \mathbf{Z}\rangle - 2\big\langle \mathbf{Z}, \mathbf{Z}_k^i\big\rangle + \big\langle \mathbf{Z}_k^i, \mathbf{Z}_k^i\big\rangle\big) \\
&= \frac{\beta + \rho_k^z}{2}\left\|\mathbf{Z} - \frac{\rho_k^z \mathbf{Z}_k^i + \beta \mathtt{vec}^{-1}\Big(\big(\mathbf{R}_k^i\big)^\top \mathbf{d}_k^i\Big)}{\beta + \rho_k^z}\right\|_F^2 + \frac{\rho_k^z}{2}\big\|\mathbf{Z}_k^i\big\|_F^2 \\
&\quad - \frac{1}{2\big(\beta + \rho_k^z\big)}\Big\|\rho_k^z \mathbf{Z}_k^i + \beta \mathtt{vec}^{-1}\Big(\big(\mathbf{R}_k^i\big)^\top \mathbf{d}_k^i\Big)\Big\|_F^2 + \frac{\beta}{2}\big\|\mathbf{R}_k^i\big\|_F^2.
\end{aligned}$$

Therefore, leaving terms independent of $\mathbf{Z}$ and adding the nuclear norm term, the minimization problem in (18) is equivalent to

$$\mathbf{Z}_{k+1}^i \in \arg\min_{\mathbf{Z}} \ \|\mathbf{Z}\|_* + \frac{\beta + \rho_k^z}{2}\big\|\mathbf{Z} - \mathbf{M}_k^i\big\|_F^2 \tag{20}$$

where $\mathbf{M}_k^i = (\rho_k^z \mathbf{Z}_k^i + \beta \mathtt{vec}^{-1}((\mathbf{R}_k^i)^\top \mathbf{d}_k^i))/(\beta + \rho_k^z)$. Then, we can directly derive the closed-form solution of (20) with the singular value thresholding (SVT) operator [55] as

$$\mathbf{Z}_{k+1}^i = \mathtt{SVT}_{\frac{1}{\beta+\rho_k^z}}\big(\mathbf{M}_k^i\big) \triangleq \mathbf{U}\left(\mathbf{S} - \frac{1}{\beta + \rho_k^z}\right)_+ \mathbf{V}^\top \tag{21}$$

where $(\mathbf{U}, \mathbf{S}, \mathbf{V})$ comes from the SVD of $\mathbf{M}_k^i$, $\mathbf{S}$ is a diagonal matrix with $\mathbf{M}^{k,i}$'s singular values, and $(\cdot)_+$ means keeping the positive values and setting the negative values as 0.

Similarly, we can obtain the closed-form solution of (19) as follows:

$$\mathbf{d}_{k+1}^i = \frac{\beta \mathbf{R}_k^i \mathtt{vec}\big(\mathbf{Z}_k^i\big) + \rho_k^d \mathbf{d}_i^k}{\big\|\beta \mathbf{R}_k^i \mathtt{vec}\big(\mathbf{Z}_k^i\big) + \rho_k^d \mathbf{d}_i^k\big\|_2}. \tag{22}$$

Afterward, we obtain $\mathcal{Z}_{k+1}$ with its ith frontal slice equaling to $\mathbf{Z}_{k+1}^i$ and $\mathbf{D}_{k+1} = [\mathbf{d}_{k+1}^1, \ldots, \mathbf{d}_{k+1}^i, \ldots, \mathbf{d}_{k+1}^d]$.

2) Updating $\mathcal{X}$: We update $\mathcal{X}$ via solving the following minimization problem:

$$\min_{\mathcal{X}} \frac{\beta}{2}\|\mathcal{X} - \mathcal{Z}_{k+1} \times_3 \mathbf{D}_{k+1}\|_F^2 + \Phi(\mathcal{X}) + \frac{\rho_k^x}{2}\|\mathcal{X} - \mathcal{X}_k\|_F^2.$$

$\mathcal{X}_{k+1}$ is updated via the following steps:

$$\begin{cases} \mathcal{X}_{k+\frac{1}{2}} = \dfrac{\beta \mathtt{fold}_3\big(\mathbf{D}_{k+1}\mathbf{Z}_{k+1(3)}\big) + \rho_k^x \mathcal{X}_k}{\beta + \rho_k^x} \\ \mathcal{X}_{k+1} = \Big(\mathcal{X}_{k+\frac{1}{2}}\Big)_{\Omega^C} + \mathcal{O}_\Omega \end{cases} \tag{23}$$

where Ω_C denotes the complementary set of the Ω. Finally, the pseudocode is summarized in Algorithm 1. The computation complexity of our algorithm at each iteration is $O(dn_1n_2(dn_3 + \min(n_1, n_2) + n_3)$, given an input with size $n_1 \times n_2 \times n_3$.

Algorithm 1 Proximal Alternating Minimization Algorithm for Solving (13)

Input: The observed tensor $\mathcal{O} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$; the set of observed entries Ω.
Initialization: $\mathcal{X}^{(0)}$, $\mathbf{D}^0$, and $\mathbf{Z}^0$;
1: **while** not converged **do**
2: **for** $i = 1$ to d **do**
3: Update $\mathcal{Z}^{k+1}(:,:,i)$ via Eq. (20);
4: Update $\mathbf{D}^{k+1}(:,i)$ via (22);
5: **end for**
6: Update $\mathcal{X}^k$ (23).
7: **end while**
Output: The reconstructed tensor $\mathcal{X}$.

C. Convergency Analysis

In this section, we establish the theoretical guarantee of convergence on our algorithm. For convenience, we first define the following formularies:

$$\begin{aligned} F(\mathcal{Z}) &= \sum_{k=1}^{d} \left\| \mathcal{Z}^{(k)} \right\|_* = \|\mathtt{bdiag}(\mathcal{Z})\|_* \\ \delta_{\mathcal{X}}(\mathcal{X}) &= \Phi(\mathcal{X}) \\ \delta_{\mathcal{D}}(\mathbf{D}) &= \Psi(\mathbf{D}) \\ Q(\mathcal{Z}, \mathbf{D}, \mathcal{X}) &= \frac{\beta}{2}\|\mathcal{X} - \mathcal{Z} \times_3 \mathbf{D}\|_F^2 \\ L(\mathcal{Z}, \mathbf{D}, \mathcal{X}) &= F(\mathcal{Z}) + \delta_{\mathcal{X}}(\mathcal{X}) + \delta_{\mathcal{D}}(\mathbf{D}) + Q(\mathcal{Z}, \mathbf{D}, \mathcal{X}) \end{aligned}$$

and

$$\begin{cases} \mathcal{Z}_{k+1} = \arg\min\limits_{\mathcal{Z}} \Big\{ M_1(\mathcal{Z}|\mathcal{Z}_k) := F(\mathcal{Z}) \\ \qquad\qquad + Q(\mathcal{Z}, \mathbf{D}_k, \mathcal{X}_k) + \frac{\rho_k^z}{2}\|\mathcal{Z} - \mathcal{Z}_k\|_F^2 \Big\} \\ \mathbf{D}_{k+1} = \arg\min\limits_{\mathbf{D}} \Big\{ M_2(\mathbf{D}|\mathbf{D}_k) := \delta_{\mathcal{X}}(\mathcal{X}) \\ \qquad\qquad + Q(\mathcal{Z}_{k+1}, \mathbf{D}, \mathcal{X}_k) + \frac{\rho_k^d}{2}\|\mathbf{D} - \mathbf{D}_k\|_F^2 \Big\} \\ \mathcal{X}_{k+1} = \arg\min\limits_{\mathcal{X}} \Big\{ M_3(\mathcal{X}|\mathcal{X}_k) := \delta_{\mathcal{D}}(\mathbf{D}) \\ \qquad\qquad + Q(\mathcal{Z}_{k+1}, \mathbf{D}_{k+1}, \mathcal{X}) + \frac{\rho_k^x}{2}\|\mathcal{X} - \mathcal{X}_k\|_F^2 \Big\}. \end{cases} \tag{24}$$

Next, we give the theorem of global convergency of the sequence generated by (24) as follows.

Theorem 2: The sequence generated by (24) is bounded, and it converges to a critical point of $L(\mathcal{Z}, \mathbf{D}, \mathcal{X})$.

As the process of updating in (24) is factually a special instance of Algorithm 4 described in [56], the proof of Theorem 2 confirms to [56, Th. 6.2] if satisfying the following conditions:

$$\begin{cases} 1) \text{ the K-Ł property of } L \text{ at each point} \\ 2) \text{ the sufficient decrease condition ((64) in [56])} \\ 3) \text{ the relative error condition ((65) and (66) in [56]).} \end{cases}$$

The road map of the proof also follows this line. Before verifying these conditions, we first give some basic definitions from variational analysis [57], [58]. If $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is a real-extended-valued function, its domain is given by $\mathrm{dom} f := \{x \in \mathbb{R}^n : f(x) < +\infty\}$. For each $x \in \mathrm{dom} f$, the *Fréchet subdifferential* of f at x, written $\hat{\partial} f(x)$, is the set of vectors $x^* \in \mathbb{R}^n$ which satisfy

$$\liminf_{y \neq x, y \to x} \frac{1}{\|x - y\|}\big[f(y) - f(x) - \langle x^*, y - x\rangle \geq 0.$$

When $x \notin \mathrm{dom} f$, we set $\hat{\partial} f(x) = \emptyset$. Then, the subdifferential (*limiting subdifferential* [57]) of f at $x \in \mathrm{dom} f$, written $\partial f(x)$, is a set defined as

$$\left\{x^* \in \mathbb{R}^n : \exists x_n \to x,\, f(x_n) \to f(x), x_n^* \in \hat{\partial} f(x_n) \to x^*\right\}.$$

The (limiting) subdifferential is more stable than the Fréchet subdifferential in an algorithmic context which involves limiting processes. A necessary (but not sufficient) condition for $x \in \mathbb{R}^n$ to be a minimizer of f is $\partial f(x) \ni 0$. A point that satisfies $\partial f(x) \ni 0$ is called limiting critical or simply critical. If K is a subset of $\mathbb{R}^n$ and x is any point in $\mathbb{R}^n$, we set

$$\mathrm{dist}(x, K) = \inf\{\|x - z\| : z \in K\}.$$

If K is empty, we have $\mathrm{dist}(x, K) = +\infty$ for all $x \in \mathbb{R}^n$. For any real-extended-valued function f on $\mathbb{R}^n$, we have $\mathrm{dist}(0, \partial f(x)) = \inf\{\|x^*\| : x^* \in \partial f(x)\}$ [59]. Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a proper lower semicontinuous function. For $-\infty < \eta_1 < \eta_2 \leq +\infty$, we set

$$[\eta_1 < f < \eta_2] = \left\{x \in \mathbb{R}^n : \eta_1 < f(x) < \eta_2\right\}.$$

Then, we can define K-Ł functions and semialgebraic functions.

Definition 9 (Kurdyka-Łojasiewicz Property [56]): A proper lower semicontinuous function $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is said to have the K-Ł property at $\bar{x} \in \mathrm{dom}(\partial f)$ if there exist $\eta \in (0, +\infty]$, a neighborhood U of $\bar{x}$ and a continuous concave function $\phi : [0, \eta) \to [0, +\infty]$, which satisfies $\phi(0) = 0$, ϕ is C^1 on $(0, \eta)$, and $\phi(s) > 0, \forall s \in (0, \eta))$ such that for each $x \in U \cap [f(\bar{x}) < f < f(\bar{x}) + \eta]$ the K-Ł inequality holds

$$\phi'(f(x) - f(\bar{x}))\mathrm{dist}(0, \partial f(x)) \geq 1. \tag{25}$$

If f satisfies the K-Ł property at each point of $\mathrm{dom}\partial f$, then f is called a K-Ł function.

Definition 10 (Semialgebraic Sets and Functions [56]): A subset S of $\mathbb{R}$ is called semialgebraic set if there exists a finite number of real polynomial functions g_{ij}, h_{ij} such that $S = \bigcap_j \bigcup_i \{x \in \mathbb{R}^n : g_{ij}(x) = 0, h_{ij}(x) < 0\}$. A function f is called semialgebraic function if its graph $\{(x, t) \in \mathbb{R}^n \times \mathbb{R}, t = f(x)\}$ is a semialgebraic set.

Next, we verify the K-Ł property of L and then show the descent Lemma for $L(\mathcal{Z}, \mathbf{D}_k, \mathcal{X}_k)$. Afterward, the relative error Lemma would be given. Finally, we establish the proof of Theorem 2.

Lemma 1 (K-Ł Property Lemma): The function L satisfies the K-Ł property at each point.

Proof of Lemma 1: It is easy to verify that Q is C^1 function with locally Lipschitz continuous gradient and F, δ_D, and $\delta_{\mathcal{X}}$ are proper and lower semicontinuous. Thus, L is a proper

lower semicontinuous function. The nuclear norm and Frobenius norm are semialgebraic [52]. Additionally, the indicator function with semialgebraic sets is semialgebraic [52]. As a semialgebraic real-valued function f is a K-Ł function, i.e., f satisfies K-Ł property at each $x \in \mathrm{dom}(f)$ [60], the function L satisfies the K-Ł property at each point. □

Lemma 2 (Descent Lemma): Assume that $L(\mathcal{Z}, \mathbf{D}, \mathcal{X})$ is a C^1 function with locally Lipschitz continuous gradient and $\rho_k^z, \rho_k^d, \rho_k^x > 0$. Let $\{\mathcal{Z}_k, \mathbf{D}_k, \mathcal{X}_k\}_{k\in\mathbb{N}}$ be generated by (24). Then

$$\begin{aligned}
&F(\mathcal{Z}_{k+1}) + Q(\mathcal{Z}_{k+1}, \mathbf{D}_k, \mathcal{X}_k) + \frac{\rho_k^z}{2}\|\mathcal{Z}_{k+1} - \mathcal{Z}_k\|_F^2 \\
&\le F(\mathcal{Z}_k) + Q(\mathcal{Z}_k, \mathbf{D}_k, \mathcal{X}_k) \\
&\delta_{\mathcal{D}}(\mathbf{D}_{k+1}) + Q(\mathcal{Z}_{k+1}, \mathbf{D}_{k+1}, \mathcal{X}_k) + \frac{\rho_k^d}{2}\|\mathbf{D}_{k+1} - \mathbf{D}_k\|_F^2 \\
&\le \delta_{\mathcal{D}}(\mathbf{D}_k) + Q(\mathcal{Z}_{k+1}, \mathbf{D}_k, \mathcal{X}_k) \\
&\delta_{\mathcal{X}}(\mathcal{X}_{k+1}) + Q(\mathcal{Z}_{k+1}, \mathbf{D}_{k+1}, \mathcal{X}_{k+1}) + \frac{\rho_k^x}{2}\|\mathcal{X}_{k+1} - \mathcal{X}_k\|_F^2 \\
&\le \delta_{\mathcal{X}}(\mathcal{X}_k) + Q(\mathcal{Z}_{k+1}, \mathbf{D}_{k+1}, \mathcal{X}_k).
\end{aligned}$$

Proof of Lemma 2: When $\mathbf{D}_{k+1}$ and $\mathcal{X}_{k+1}$ are optimal solutions of M_2 and M_3, $\delta_{\mathcal{D}} = 0$ and $\delta_{\mathcal{X}} = 0$. By the definitions of M_1–M_3, we clearly have that

$$\begin{aligned}
&F(\mathcal{Z}_{k+1}) + Q(\mathcal{Z}_{k+1}, \mathbf{D}_k, \mathcal{X}_k) + \frac{\rho_k^z}{2}\|\mathcal{Z}_{k+1} - \mathcal{Z}_k\|_F^2 \\
&= M_1(\mathcal{Z}_{k+1}|\mathcal{Z}_k) \le M_1(\mathcal{Z}_k|\mathcal{Z}_k) \\
&= F(\mathcal{Z}_k) + Q(\mathcal{Z}_k, \mathbf{D}_k, \mathcal{X}_k) \\
&\delta_{\mathcal{D}}(\mathbf{D}_{k+1}) + Q(\mathcal{Z}_{k+1}, \mathbf{D}_{k+1}, \mathcal{X}_k) + \frac{\rho_k^d}{2}\|\mathbf{D}_{k+1} - \mathbf{D}_k\|_F^2 \\
&= M_2(\mathbf{D}_{k+1}|\mathbf{D}_k) \le M_2(\mathbf{D}_k|\mathbf{D}_k) \\
&= \delta_{\mathcal{D}}(\mathbf{D}_k) + Q(\mathcal{Z}_{k+1}, \mathbf{D}_k, \mathcal{X}_k) \\
&\delta_{\mathcal{X}}(\mathcal{X}_{k+1}) + Q(\mathcal{Z}_{k+1}, \mathbf{D}_{k+1}, \mathcal{X}_{k+1}) + \frac{\rho_k^x}{2}\|\mathcal{X}_{k+1} - \mathcal{X}_k\|_F^2 \\
&= M_3(\mathcal{X}_{k+1}|\mathcal{X}_k) \le M_3(\mathcal{X}_k|\mathcal{X}_k) \\
&= \delta_{\mathcal{X}}(\mathcal{X}_k) + Q(\mathcal{Z}_{k+1}, \mathbf{D}_{k+1}, \mathcal{X}_k).
\end{aligned}$$

The descent lemma has been proven. □

Lemma 3 (Relative Error Lemma): $\{\mathcal{Z}_k, \mathbf{D}_k, \mathcal{X}_k\}_{k\in\mathbb{N}}$ is generated by (24) and $\rho_k^z, \rho_k^d, \rho_k^x > 0$. Then there exists $V_{1,k+1}, V_{2,k+1}, V_{3,k+1}$, which satisfy the following formularies:

$$\begin{aligned}
\left\|V_{k+1}^1 + \nabla_{\mathcal{Z}} Q(\mathcal{Z}_{k+1}, \mathbf{D}_k, \mathcal{X}_k)\right\|_F &\le \rho_k^z\|\mathcal{Z}_{k+1} - \mathcal{Z}_k\|_F \\
\left\|V_{k+1}^2 + \nabla_{\mathbf{D}} Q(\mathcal{Z}_{k+1}, \mathbf{D}_{k+1}, \mathcal{X}_k)\right\|_F &\le \rho_k^d\|\mathbf{D}_{k+1} - \mathbf{D}_k\|_F \\
\left\|V_{k+1}^3 + \nabla_{\mathcal{X}} Q(\mathcal{Z}_{k+1}, \mathbf{D}_{k+1}, \mathcal{X}_{k+1})\right\|_F &\le \rho_k^x\|\mathcal{X}_{k+1} - \mathcal{X}_k\|_F
\end{aligned}$$

where $V_{k+1}^1 \in \partial F(\mathcal{Z}_{k+1})$, $V_{k+1}^2 \in \partial\delta_{\mathcal{D}}(\mathbf{D}_{k+1})$, $V_{k+1}^3 \in \partial\delta_{\mathcal{X}}(\mathcal{X}_{k+1})$, and ∇ indicates the (partial) gradient.

Proof of Lemma 3: By the definition of M_1–M_3, we have

$$\begin{aligned}
0 &\in \partial F(\mathcal{Z}_{k+1}) + \nabla_{\mathcal{Z}} Q(\mathcal{Z}_{k+1}, \mathbf{D}_k, \mathcal{X}_k) + \rho_k^z(\mathcal{Z}_{k+1} - \mathcal{Z}_k) \\
0 &\in \partial\delta_{\mathcal{D}}(\mathbf{D}_{k+1}) + \nabla_{\mathbf{D}} Q(\mathcal{Z}_{k+1}, \mathbf{D}_{k+1}, \mathcal{X}_k) + \rho_k^d(\mathbf{D}_{k+1} - \mathbf{D}_k) \\
0 &\in \partial\delta_{\mathcal{X}}(\mathcal{X}_{k+1}) + \nabla_{\mathcal{X}} Q(\mathcal{Z}_{k+1}, \mathbf{D}_{k+1}, \mathcal{X}_{k+1}) + \rho_k^x(\mathcal{X}_{k+1} - \mathcal{X}_k).
\end{aligned}$$

Let

$$\begin{cases}
V_{k+1}^1 := -\nabla_{\mathcal{Z}} Q(\mathcal{Z}_{k+1}, \mathbf{D}_k, \mathcal{X}_k) - \rho_k^z(\mathcal{Z}_{k+1} - \mathcal{Z}_k) \\
V_{k+1}^2 := -\nabla_{\mathcal{Z}} Q(\mathcal{Z}_{k+1}, \mathbf{D}_{k+1}, \mathcal{X}_k) - \rho_k^d(\mathbf{D}_{k+1} - \mathbf{D}_k) \\
V_{k+1}^3 := -\nabla_{\mathcal{Z}} Q(\mathcal{Z}_{k+1}, \mathbf{D}_{k+1}, \mathcal{X}_k) - \rho_k^x(\mathcal{X}_{k+1} - \mathcal{X}_k).
\end{cases}$$

It is clear that $V_{k+1}^1 \in \partial F(\mathcal{Z}_{k+1})$, $V_{k+1}^2 \in \partial\delta_{\mathcal{D}}(\mathbf{D}_{k+1})$, and $V_{k+1}^3 \in \partial\delta_{\mathcal{X}}(\mathcal{X}_{k+1})$. Thus, we have

$$\begin{cases}
\left\|V_{k+1}^1 + \nabla_{\mathcal{Z}} Q(\mathcal{Z}_{k+1}, \mathbf{D}_k, \mathcal{X}_k)\right\|_F = \rho_k^z\|\mathcal{Z}_{k+1} - \mathcal{Z}_k\|_F \\
\left\|V_{k+1}^2 + \nabla_{\mathbf{D}} Q(\mathcal{Z}_{k+1}, \mathbf{D}_{k+1}, \mathcal{X}_k)\right\|_F = \rho_k^d\|\mathbf{D}_{k+1} - \mathbf{D}_k\|_F \\
\left\|V_{k+1}^3 + \nabla_{\mathcal{X}} Q(\mathcal{Z}_{k+1}, \mathbf{D}_{k+1}, \mathcal{X}_{k+1})\right\|_F = \rho_k^x\|\mathcal{X}_{k+1} - \mathcal{X}_k\|_F.
\end{cases}$$

The proof of relative error Lemma has been finished. □

Now, we begin to establish our proof of Theorem 2.

Proof of Theorem 2: From Lemma 2, we have that the objective function value monotonically decreases. First, we can see that the indicator function $\delta_{\mathcal{D}}(\mathbf{D}) = \Phi(\mathbf{D})$ should be 0 from its definition. Thus,

$$\|\mathbf{D}\|_F^2 = \sum_i \|\mathbf{D}(:, i)\|_2^2 = d$$

which means $\{\mathbf{D}_k\}_{k\in\mathbb{N}}$ *is bounded*. Meanwhile, from the monotonic decreasing, the nonnegative terms $F(\mathcal{Z}) = \sum_{k=1}^d \|\mathcal{Z}^{(k)}\|_*$ and $Q(\mathcal{Z}, \mathbf{D}, \mathcal{X}) = (\beta/2)\|\mathcal{X} - \mathcal{Z} \times_3 \mathbf{D}\|_F^2$ are bounded. Then

$$\|\mathcal{Z}\|_F^2 = \sum_{k=1}^d \left\|\mathcal{Z}^{(k)}\right\|_F^2 \le \sum_{k=1}^d \left\|\left(\mathcal{Z}^{(k)}\right)\right\|_*^2.$$

That is, $\{\mathbf{Z}_k\}_{k\in\mathbb{N}}$ is bounded. Next, from the triangle inequality, we have

$$\|\mathcal{X}\|_F - \|\mathcal{Z}\|_F\|\mathbf{D}\|_F \le \|\mathcal{X}\|_F - \|\mathcal{Z} \times_3 \mathbf{D}\|_F \le \|\mathcal{X} - \mathcal{Z} \times_3 \mathbf{D}\|_F.$$

Therefore, $\{\mathbf{X}_k\}_{k\in\mathbb{N}}$ is bounded.

By Lemma 1, the sequence $\{\mathcal{Z}_k, \mathbf{D}_k, \mathcal{X}_k\}_{k\in\mathbb{N}}$ is a bounded sequence with the K-Ł property at each point. Combining Lemmas 2 and 3 with the above property of L, the process of updating in (24) is factually a special instance of Algorithm 4 described in [56]. Lemmas 2 and 3 correspond to [56, Eqs. (64)–(66)]. Under these conditions, this proof conforms to [56, Th. 6.2]. Thus, the bounded sequence $\{\mathcal{Z}_k, \mathbf{D}_k, \mathcal{X}_k\}_{k\in\mathbb{N}}$ converges to a critical point of $L(\mathcal{Z}, \mathbf{D}, \mathcal{X})$. □

Algorithm 1 is a direct multiblock generalization of (24). The proof of its convergence accords with the proof of Theorem 2 and can be easily obtained. Meanwhile, the above convergence analysis is more similar to the analysis in [56], being convenient for the verification of readers. Therefore, we establish the proof of Theorem 2 here.

IV. Numerical Experiments

In this section, we compare our method with other state-of-the-art methods. The compared methods consist of: one baseline Tucker-rank-based method HaLRTC[1] [4], a Bayesian CP-factorization-based method (BCPF[2]) [14], a TR-decomposition-based method (TRLRF[3]) [24], a t-SVD-based method (TNN[4]) [33], a DCT-induced TNN minimization method (DCTNN[5]) [37], and a framelet-represented TNN

[1]https://www.cs.rochester.edu/jliu/code/TensorCompletion.zip

[2]https://github.com/qbzhao/BCPF

[3]https://github.com/yuanlonghao/TRLRF

[4]https://github.com/jamiezeminzhang/Tensor_Completion_and_Tensor_RPCA

[5]Implemented by ourselves based on the code of TNN.

TABLE I

PSNR, SSIM, AND UIQI OF RESULTS BY DIFFERENT METHODS WITH DIFFERENT SRS ON THE **VIDEO** DATA. THE BEST, THE SECOND BEST, AND THE THIRD BEST VALUES ARE, RESPECTIVELY, HIGHLIGHTED BY RED, BLUE, AND GREEN COLORS

Video	SR	10%			20%			30%			40%			50%			Time
	Method	PSNR	SSIM	UIQI	PSNR	SSIM	UIQI	PSNR	SSIM	UIQI	PSNR	SSIM	UIQI	PSNR	SSIM	UIQI	(s)
foreman	Observed	3.96	0.010	0.006	4.48	0.017	0.016	5.05	0.025	0.027	5.72	0.035	0.040	6.51	0.047	0.056	0
	HaLRTC	20.10	0.511	0.328	23.88	0.700	0.562	26.77	0.812	0.699	29.35	0.883	0.790	31.85	0.928	0.853	4
	BCPF	23.58	0.610	0.458	26.03	0.723	0.586	27.27	0.769	0.638	27.98	0.793	0.666	28.42	0.808	0.682	829
	TRLRF	24.63	0.616	0.536	27.72	0.778	0.667	29.01	0.830	0.719	29.94	0.861	0.756	30.95	0.887	0.788	387
	TNN	23.71	0.606	0.489	26.90	0.748	0.640	29.13	0.824	0.720	31.28	0.880	0.783	33.47	0.921	0.835	30
	DCTNN	24.27	0.632	0.519	27.23	0.766	0.657	29.54	0.843	0.740	31.80	0.897	0.804	34.07	0.934	0.854	21
	FTNN	25.41	0.735	0.603	28.51	0.849	0.741	30.92	0.904	0.815	33.17	0.937	0.862	35.44	0.960	0.900	112
	DTNN	26.22	0.799	0.676	29.26	0.875	0.778	31.77	0.917	0.837	34.13	0.946	0.879	36.32	0.964	0.907	301
carphone	Observed	7.04	0.023	0.010	7.56	0.039	0.026	8.13	0.057	0.046	8.81	0.077	0.070	9.59	0.100	0.097	0
	HaLRTC	24.71	0.779	0.596	28.57	0.883	0.751	31.31	0.930	0.827	33.67	0.956	0.875	35.84	0.972	0.908	6
	BCPF	27.91	0.812	0.638	29.82	0.864	0.703	31.30	0.894	0.744	32.25	0.911	0.767	32.88	0.921	0.781	825
	TRLRF	29.44	0.841	0.691	32.10	0.905	0.767	33.58	0.930	0.805	34.71	0.945	0.832	35.63	0.955	0.853	414
	TNN	27.44	0.804	0.639	30.00	0.873	0.725	31.81	0.909	0.774	33.44	0.934	0.813	35.06	0.952	0.846	31
	DCTNN	28.21	0.829	0.669	30.64	0.889	0.747	32.37	0.920	0.792	33.96	0.943	0.829	35.57	0.959	0.859	21
	FTNN	29.16	0.880	0.740	31.58	0.927	0.815	33.43	0.949	0.855	35.03	0.963	0.884	36.61	0.973	0.906	118
	DTNN	29.46	0.896	0.763	32.89	0.943	0.838	35.49	0.964	0.879	37.57	0.975	0.904	39.35	0.982	0.922	275
container	Observed	4.87	0.011	0.007	5.38	0.021	0.018	5.96	0.032	0.031	6.63	0.045	0.047	7.42	0.060	0.064	0
	HaLRTC	25.96	0.855	0.614	30.58	0.935	0.779	34.98	0.970	0.879	39.56	0.986	0.937	44.54	0.994	0.969	6
	BCPF	28.97	0.880	0.629	33.38	0.926	0.707	35.77	0.944	0.748	37.81	0.954	0.777	38.66	0.960	0.795	875
	TRLRF	32.82	0.932	0.738	37.55	0.961	0.817	39.57	0.969	0.851	40.64	0.974	0.875	41.94	0.980	0.900	387
	TNN	30.04	0.909	0.715	35.55	0.963	0.853	38.81	0.979	0.905	41.55	0.986	0.934	44.01	0.991	0.954	29
	DCTNN	31.61	0.930	0.762	38.27	0.977	0.892	42.69	0.989	0.940	45.91	0.993	0.960	48.43	0.996	0.972	19
	FTNN	32.43	0.948	0.809	37.38	0.978	0.904	41.41	0.988	0.946	44.42	0.992	0.958	47.16	0.994	0.971	146
	DTNN	32.50	0.956	0.840	39.12	0.985	0.929	43.72	0.992	0.959	47.28	0.995	0.972	49.82	0.997	0.980	355
highway	Observed	3.52	0.010	0.003	4.04	0.015	0.008	4.61	0.020	0.014	5.28	0.027	0.021	6.07	0.034	0.029	0
	HaLRTC	28.80	0.854	0.604	31.55	0.909	0.730	33.57	0.937	0.797	35.26	0.954	0.844	36.83	0.966	0.880	5
	BCPF	29.96	0.840	0.593	32.10	0.879	0.664	33.17	0.897	0.693	34.16	0.912	0.716	34.57	0.918	0.729	714
	TRLRF	31.31	0.857	0.661	33.81	0.905	0.729	35.35	0.929	0.773	36.40	0.943	0.806	37.59	0.956	0.841	387
	TNN	30.19	0.852	0.631	32.07	0.893	0.704	33.57	0.917	0.753	34.90	0.936	0.794	36.26	0.951	0.832	26
	DCTNN	30.59	0.864	0.648	32.35	0.899	0.715	33.79	0.922	0.762	35.12	0.939	0.802	36.49	0.954	0.838	19
	FTNN	31.23	0.893	0.693	33.09	0.926	0.764	34.63	0.944	0.808	35.93	0.956	0.842	37.25	0.966	0.875	123
	DTNN	31.70	0.902	0.711	33.96	0.931	0.774	35.81	0.948	0.815	37.34	0.959	0.849	38.71	0.968	0.877	223

minimization method (FTNN[6]) [40]. We select four types of tensor data, including videos, HSIs, traffic data, and MRI data, to show that our method is adaptive to different types of data.

Since the algorithm of our method is a nonconvex approach, the initialization of our algorithm is important. We use a simple linear interpolation strategy, which is used in [61] and convenient to implement with low cost, to fill in the missing pixels and obtaining $\mathcal{X}_0$ for our method. As the index of the observed entries $\mathbf{\Omega}$ is known, we first sort n_1n_2 tubes of $\mathcal{X}_0 \in \mathbb{R}^{n_1\times n_2\times n_3}$ based on the number of observed entries in each tube. Then, we select the first d tubes, which contain the observed entries as much as possible to construct $\mathbf{D} \in \mathbb{R}^{d\times n_3}$. Finally, the columns of $\mathbf{D}$ are normalized to satisfy $\|\mathbf{D}(:,i)\|_2 = 1$ for $i = 1,\ldots,d$. This strategy comes from many traditionary dictionary learning techniques, such as [44]. Then, we fix $\mathcal{X} = \mathcal{X}_0$ and run ten iterations of our method to initialize $\mathcal{Z}_0$ with random inputs.

Throughout all the experiments in this article, the parameters of the proposed method are set as: $d = 5n_3$, $\beta = 10$, $\rho^z = 20$, $\rho^d = 1$, and $\rho^x = 1$. In the framework of the HQS algorithm, the penalty parameter β is required to reach infinite when iteration goes on. Therefore, we enlarge β at the 15th, 20th, and 25th iterations by multiplying the factor 1.5 and enlarge β by multiplying the factor 1.2 at each iteration from the 30th iteration until satisfying the condition of convergence. ρ^z, ρ^d, and ρ^x are selected by grid search from the candidate set $\{0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100\}$, while d and ρ are manually tuned.

As for the compared methods, their parameters are manually tuned for best performances. Specifically, as the models of TNN and DCTNN are optimized by ADMM, we set the parameter β, which is introduced when building the argument Lagrangian function, as 10^{-2} at the beginning and enlarge it with a factor 1.2 at each iteration. For other methods, we set: 1) $\alpha = [1, 1, 5]$ and $\rho = 10^{-2}$ (as referred to [4, eq. (42)]) for HaLRTC; 2) removing unnecessary components automatically, random initializations, and the initial rank 200 for BCPF; 3) TR-rank $= 12$, $\mu = 1$, and $\lambda = 10$ (as referred to [21, eq. (15)]) for TRLRF; and 4) using the default setting in [36] for FTNN.

A. Video Data

In this section, we test our method for video data completion and select four videos[7] named "*foreman*" "*carphone*" "*highway*" and "*container*" of size $144 \times 176 \times 50$ (height $\times$ width $\times$ frame) to conduct comparisons. The SR varies from 10% to 50%. We compute the peak signal-to-noise ratio (PSNR), the structural similarity (SSIM) index [62], and the universal image quality index (UIQI) [63] of the results by different methods. Higher values of these three quality metrics indicate better completion performances.

In Table I, we report the quantitative metrics of the results obtained by different methods and the average running time on the video data. From Table I, it can be found that the results by TRLRF are promising when the SR is low. The performance

[6]https://github.com/TaiXiangJiang/Framelet-TNN

[7]Videos available at http://trace.eas.asu.edu/yuv/.

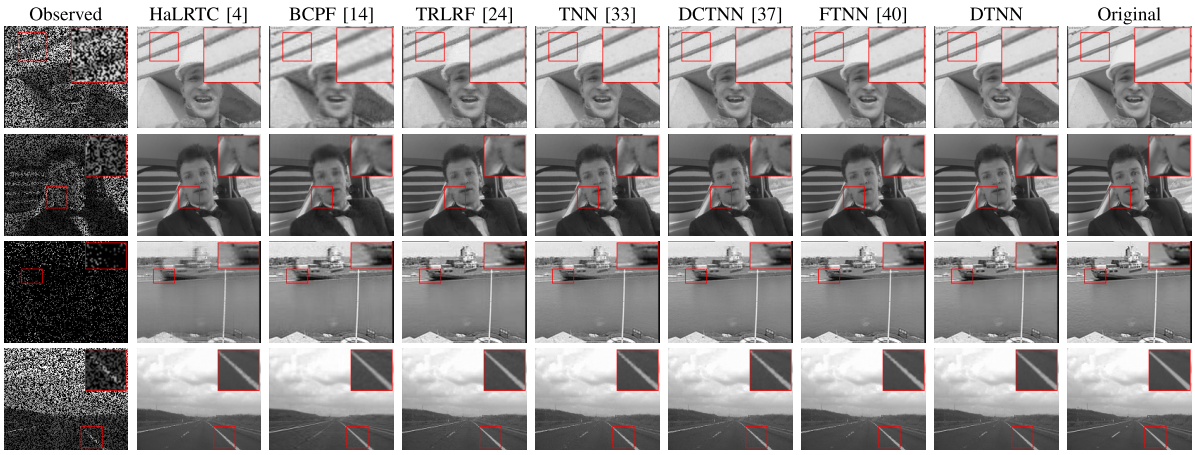


Fig. 2. One frame of the results on the **video** data. From top to bottom: the 22th frame of "foreman" (SR = 50%), the fifth frame of "carphone" (SR = 50%), the 39th frame of "container" (SR = 10%), and the 48th frame of "highway" (SR = 50%).

TABLE II

PSNR, SSIM, AND SAM OF RESULTS BY DIFFERENT METHODS WITH DIFFERENT SRS ON THE **HSI** DATA. THE BEST, THE SECOND BEST, AND THE THIRD BEST VALUES ARE, RESPECTIVELY, HIGHLIGHTED BY RED, BLUE, AND GREEN COLORS

HSI	SR	5%			10%			20%			30%			40%			50%			Time
	Method	PSNR	SSIM	SAD	PSNR	SSIM	SAD	PSNR	SSIM	SAD	PSNR	SSIM	SAD	PSNR	SSIM	SAD	PSNR	SSIM	SAD	(s)
Pavia City	Observed	12.19	0.020	1.355	12.43	0.036	1.254	12.94	0.070	1.110	13.52	0.108	0.993	14.18	0.149	0.887	14.98	0.196	0.785	0
	HaLRTC	22.95	0.596	0.126	27.67	0.835	0.095	35.68	0.970	0.048	43.11	0.994	0.024	51.84	0.999	0.010	56.13	1.000	0.006	24
	BCPF	29.12	0.850	0.100	33.00	0.927	0.077	37.71	0.970	0.051	39.51	0.980	0.043	40.12	0.982	0.041	40.78	0.985	0.038	3149
	TRLRF	33.75	0.936	0.068	37.03	0.967	0.051	38.97	0.978	0.044	40.13	0.983	0.040	41.10	0.986	0.036	42.01	0.989	0.033	1153
	TNN	26.08	0.739	0.147	32.49	0.918	0.099	38.18	0.968	0.064	41.89	0.982	0.048	45.01	0.989	0.037	48.13	0.993	0.029	97
	DCTNN	29.72	0.870	0.098	38.26	0.978	0.042	48.54	0.998	0.015	54.79	0.999	0.008	59.20	1.000	0.005	62.97	1.000	0.004	66
	FTNN	33.51	0.936	0.076	38.60	0.974	0.053	45.37	0.991	0.033	49.73	0.995	0.024	54.99	0.997	0.017	57.73	0.998	0.013	431
	DTNN	34.26	0.953	0.052	40.86	0.989	0.026	53.20	0.999	0.008	65.00	1.000	0.002	67.15	1.000	0.002	77.16	1.000	0.001	962
Washington DC	Observed	12.45	0.028	1.353	12.68	0.053	1.254	13.19	0.108	1.110	13.77	0.169	0.993	14.44	0.234	0.887	15.23	0.304	0.785	0
	HaLRTC	23.24	0.713	0.208	29.36	0.906	0.125	38.21	0.983	0.062	44.76	0.996	0.034	49.67	0.999	0.020	53.12	0.999	0.015	49
	BCPF	28.65	0.877	0.155	32.06	0.939	0.118	34.75	0.964	0.093	35.70	0.971	0.086	35.98	0.972	0.083	36.03	0.973	0.083	5566
	TRLRF	31.74	0.934	0.121	33.64	0.955	0.102	34.98	0.966	0.091	35.86	0.972	0.084	36.73	0.976	0.078	37.69	0.981	0.071	1931
	TNN	22.34	0.657	0.260	30.19	0.915	0.142	36.56	0.974	0.085	40.10	0.987	0.062	43.03	0.992	0.047	45.68	0.995	0.036	159
	DCTNN	27.09	0.840	0.182	32.59	0.946	0.116	38.16	0.982	0.072	41.85	0.992	0.051	44.86	0.995	0.038	47.50	0.997	0.029	110
	FTNN	32.17	0.946	0.107	37.03	0.979	0.075	42.96	0.992	0.048	46.64	0.996	0.036	49.46	0.997	0.028	52.06	0.998	0.021	805
	DTNN	34.21	0.969	0.075	39.85	0.992	0.044	45.01	0.997	0.033	47.56	0.998	0.029	53.35	0.999	0.015	58.59	1.000	0.009	1837

of FTNN is better than TNN and DCTNN for the video "*foreman*," while DCTNN exceeds FTNN and TNN for the video "*container*." This reveals that the predefined transformations lack flexibility. Meanwhile, with minor exceptions, our DTNN achieves the best performance for different SRs, illustrating the superiority of the data-adaptive dictionary.

Fig. 2 exhibits one frame of the results by different methods on the video data. From the enlarged area, it can be found that our DTNN well restores edges in "*foreman*" and "*highway*," the hair in "*carphone*," and the ship's outline in "*container*." The homogeneous areas are also protected by our method. We can conclude that the visual effect of our method is the best.

B. HSIs

In this section, two HSIs, i.e., a subimage of Pavia City Center dataset[8] of the size 200 × 200 × 80 (height × width × band), and a subimage of Washington DC Mall dataset[9] of the size 256 × 256 × 160, are adopted as the testing data. Since the redundancy between HSIs' slices is so high that all the methods perform very well with SR = 50%, we add the case with SR = 5%. Thus, the SRs vary from 5% to 50%. Three numerical metrics, consisting of PSNR, SSIM, and the mean spectral angle mapper (SAM) [64], are selected to quantitatively measure the reconstructed results. Lower values of SAM indicate better reconstructions.

In Table II, we show the quantitative comparisons of different methods on HSIs. FTNN and TRLRF perform well for low SR. We can also see that DCTNN and FTNN alternatively achieve the second best place in many cases, showing that DCT and framelet transformation fit the HSI data better than DFT. For different metrics, our DTNN obtains the best values in all cases. As SRs arise, the superiority of our method over the compared methods is more evident. For example, when dealing with Pavia City Center, the margins are at least 7.95 and 14.19 dB for PSNR when SR is 40% and 50%, respectively. We attribute this to the fact that the dictionary could be learned with better ability to express the data when the SR is high.

We display the pseudocolor images (using three bands to compose the RGB channels) of the reconstructed HSIs in Fig. 3. The similarity of the color reflects the fidelity along the spectral direction, which is of vital importance in applications of HSIs. It can be found that color distortion

[8]http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes

[9]https://engineering.purdue.edu/biehl/MultiSpec/hyperspectral.html

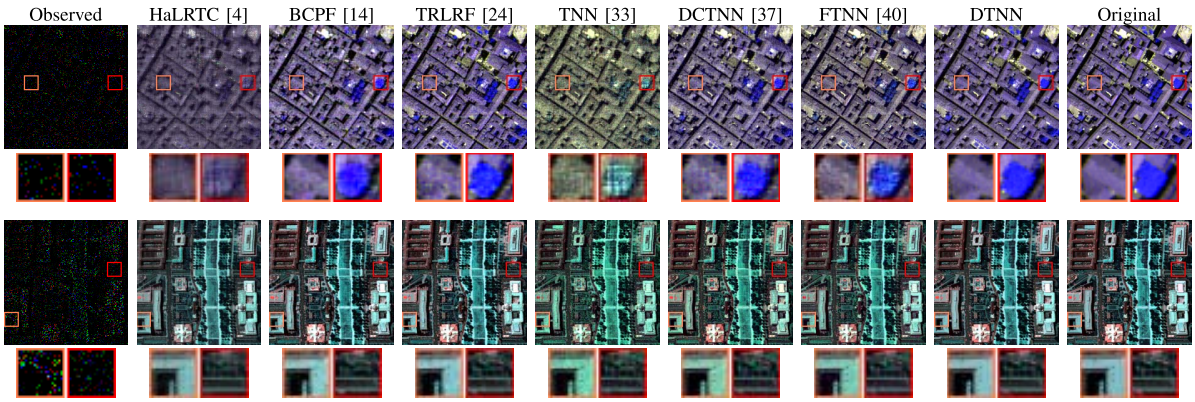


Fig. 3. Pseudocolor images and the corresponding enlarged areas of the results by different methods. Top: Pavia City Center (R-4 G-12 B-68) with SR = 5%. Bottom: Washington DC Mall (R-1 G-113 B-116) with SR = 10%.

TABLE III

RMSE AND MAPE OF RESULTS BY DIFFERENT METHODS WITH DIFFERENT SRS ON THE **TRAFFIC** DATA. THE BEST, THE SECOND BEST, AND THE THIRD BEST VALUES ARE, RESPECTIVELY, HIGHLIGHTED BY RED, BLUE, AND GREEN COLORS

SR	5%		10%		15%		20%		25%		30%		Time
Method	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	(s)
Observed	0.9148	95.71 %	0.8942	91.44 %	0.8731	87.15 %	0.8514	82.87 %	0.8291	78.56 %	0.8061	74.26 %	0
HaLRTC	0.3592	17.51 %	0.3581	16.72 %	0.3575	16.26 %	0.3571	15.94 %	0.3569	15.68 %	0.3566	15.47 %	13
BCPF	0.3594	17.25 %	0.3576	16.45 %	0.3571	16.13 %	0.3568	15.97 %	0.3567	15.84 %	0.3566	15.76 %	218
TRLRF	0.1781	9.09 %	0.1753	8.62 %	0.1749	8.42 %	0.1747	8.29 %	0.1745	8.20 %	0.1744	8.08 %	107
TNN	0.0509	3.63 %	0.0387	2.75 %	0.0336	2.33 %	0.0304	2.03 %	0.0283	1.82 %	0.0267	1.65 %	27
DCTNN	0.0480	3.38 %	0.0387	2.71 %	0.0338	2.30 %	0.0315	2.07 %	0.0296	1.88 %	0.0278	1.70 %	18
FTNN	0.0452	3.33 %	0.0358	2.31 %	0.0316	1.96 %	0.0287	1.69 %	0.0265	1.49 %	0.0247	1.32 %	129
DTNN	0.0428	2.76 %	0.0354	2.19 %	0.0305	1.84 %	0.0278	1.60 %	0.0256	1.42 %	0.0237	1.26 %	264

occurs in the results by TNN. From the enlarged orange and red boxes, we can see that DTNN outperforms the compared methods considering the spatial structures and details.

C. Traffic Data

In this section, we test all the methods on the traffic data,[10] which is provided by Grenoble Traffic Lab (GTL). A set of traffic speed data of 207 days (April 1, 2015–October 24, 2015), 1440 time windows,[11] and 21 detection points are downloaded and constitute a third-order tensor of size $1440 \times 207 \times 21$. To reduce the time consumption, a subset of the data with size $400 \times 200 \times 21$ corresponding to the first continuous 400 time windows in a day and the first 200 days is manually clipped as the ground-truth complete testing data. We select the root mean square error (RMSE)[12] and the mean absolute percentage error (MAPE)[13] to quantitatively measure the quality of the results. Lower values of RMSE and MAPE indicate better reconstructions. After random sampling the elements with SR $\in \{5\%, 10\%, 15\%, \ldots, 30\%\}$, 3 adjacent frontal slices in a random location are set as unobserved. This is to simulate the situations in which some detectors are broken. The 200th lateral slice of the observation is shown in the top left of Fig. 4, the missing slices corresponding to the blue columns.

Table III gives the quantitative metrics of the results by different methods with different SRs. We can find that the capabilities of HaLRTC, BCPF, and TRLRF are limited and this phenomenon accords with the visual results shown in Fig. 4. The effectiveness of these three methods is severely affected due to the missing frontal slices. TNN and DCTNN get better metrics while their performance is also not well considering the location of missing slices. FTNN and DTNN recover the rough structure of the missing slices and the metrics of their results also achieve the best and the second best places. The reconstruction of our DTNN in the area of missing slices is closer to the original data than FTNN.

D. MRI Data

In this section, all the methods are conducted on the MRI data[14] of size $142 \times 178 \times 121$. These MRI data provide a 3-D view of the brain part of a human being. That is, all the modes of these MRI data are corresponding to spatial information. The SRs are set from 10% to 50%. Similar to the video data, we compute the mean values of PSNR, SSIM, and UIQI of each frontal slice and report them in Table IV. From Table IV, we can find that DTNN outperforms the compared methods while DCTNN and FTNN alternatively obtain the second best values. Fig. 5 presents the 61st frontal slice of the results by different methods. For the enlarged white manner area, which is smooth, the results by our DTNN are the cleanest compared with the results by other methods.

[10] https://gtl.inrialpes.fr/data_download

[11] The sampling period is 1 min, so there are $60 \times 24 = 1440$ time windows for each day.

[12] $\text{RMSE} = ((\sum_{ijk}(\mathcal{X}^{\text{Rec}}_{ijk} - \mathcal{X}^{\text{GT}}_{ijk})^2)/n_1\ n_2\ n_3)^{1/2}$.

[13] $\text{MAPE} = 1/(n_1 n_2 n_3) \sum_{ijk}(|\mathcal{X}^{\text{Rec}}_{ijk} - \mathcal{X}^{\text{GT}}_{ijk}|/\mathcal{X}^{\text{GT}}_{ijk}) \times 100\%$. This index is a measure of prediction accuracy, usually expressing accuracy as percentage.

[14] https://brainweb.bic.mni.mcgill.ca/brainweb/selection_normal.html

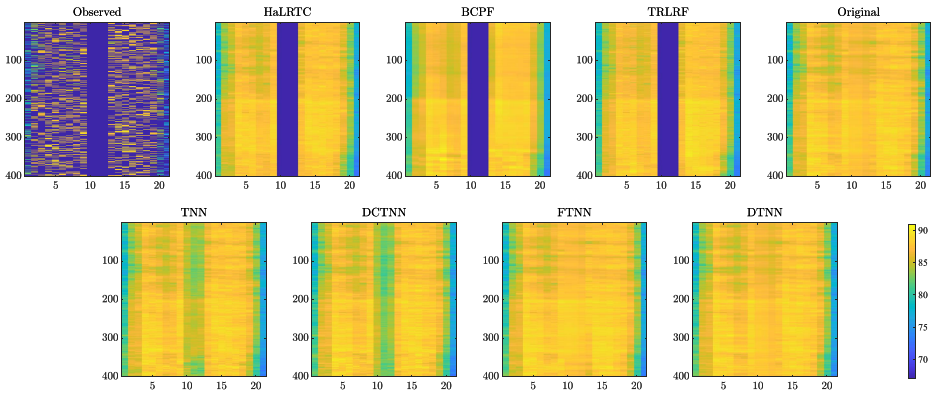


Fig. 4. 88th lateral slice of the reconstructions by different methods on the **traffic** data (SR = 30%).

TABLE IV

PSNR, SSIM, AND UIQI OF RESULTS BY DIFFERENT METHODS WITH DIFFERENT SRS ON THE **MRI** DATA. THE BEST, THE SECOND BEST, AND THE THIRD BEST VALUES ARE, RESPECTIVELY, HIGHLIGHTED BY RED, BLUE, AND GREEN COLORS

SR	10%			20%			30%			40%			50%			Time
Method	PSNR	SSIM	UIQI	PSNR	SSIM	UIQI	PSNR	SSIM	UIQI	PSNR	SSIM	UIQI	PSNR	SSIM	UIQI	(s)
Observed	8.09	0.043	0.020	8.60	0.070	0.050	9.18	0.099	0.086	9.85	0.132	0.127	10.64	0.167	0.173	0
HaLRTC	18.34	0.436	0.349	22.48	0.651	0.606	26.01	0.794	0.756	29.19	0.880	0.840	32.13	0.931	0.889	11
TRLRF	23.89	0.637	0.606	25.48	0.720	0.682	26.36	0.760	0.722	27.17	0.794	0.756	28.06	0.825	0.786	921
BCPF	22.63	0.574	0.518	24.70	0.678	0.631	25.37	0.710	0.663	25.55	0.720	0.671	25.71	0.727	0.678	2430
TNN	22.41	0.577	0.550	27.12	0.789	0.757	30.01	0.874	0.833	32.55	0.922	0.876	35.01	0.953	0.905	60
DCTNN	23.79	0.644	0.617	27.63	0.808	0.773	30.56	0.888	0.844	33.15	0.932	0.885	35.62	0.960	0.911	45
FTNN	25.15	0.743	0.695	29.02	0.872	0.825	31.96	0.928	0.883	34.49	0.958	0.916	36.89	0.975	0.937	253
DTNN	27.19	0.835	0.790	31.03	0.917	0.870	33.65	0.949	0.902	35.83	0.966	0.920	37.91	0.978	0.934	568

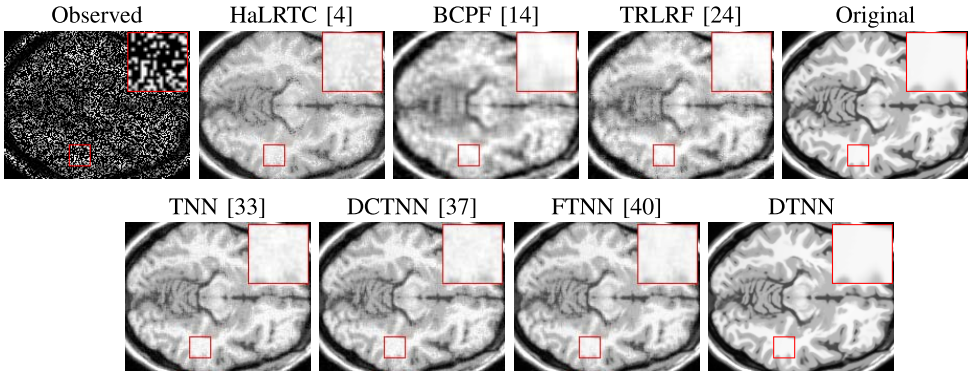


Fig. 5. 61st frontal slice of the results on the **MRI** data by different methods (SR = 30%).

E. Discussions

1) Comparisons With Traditional Dictionary Learning and Sparse Coding Approaches: In this section, we compare our method with traditional dictionary methods. First, the data $\mathcal{O} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and the coefficient $\mathcal{Z} \in \mathbb{R}^{n_1 \times n_2 \times d}$ in the tensor format are reshaped into the matrix form via the $\texttt{unfold}_3$ operation, i.e., $\mathbf{O}_{(3)} \in \mathbb{R}^{n_3 \times n_1 n_2} = \texttt{unfold}_3(\mathcal{O})$ and $\mathbf{Z}_{(3)} \in \mathbb{R}^{d \times n_1 n_2} = \texttt{unfold}_3(\mathcal{Z})$. The representation formulary also turns from $\mathcal{O} \approx \mathcal{Z} \times_3 \mathbf{D}$ to $\mathbf{O}_{(3)} \approx \mathbf{D}\mathbf{Z}_{(3)}$. The tubes of $\mathcal{Z}$ constitute the columns of $\mathbf{Z}_{(3)}$, and the ith row of $\mathbf{Z}_{(3)}$ is reshaped from the ith frontal slice of $\mathcal{Z}$. Then, for the coding coefficient matrix $\mathbf{Z}_{(3)}$ (also denoted as $\mathbf{Z}$ for convenience), we regularize it with the common $\|\mathbf{Z}\|_1 = \sum_{ij} |Z_{(3)}|$, which is usually used to enhance the sparsity, and $\|\mathbf{Z}\|_{1,2} = \sum_j (\sum_i Z_{ij}^2)^{1/2}$, which could exploit the group sparsity of the columns. The $\ell_{1,2}$ norm of $\mathbf{Z}_{(3)}$ is also mathematically equivalent to the tubal sparsity regularization, defined as $\|\mathcal{Z}\|_{1,1,2} = \sum_{i,j} \|\mathcal{Z}(i, j, :)\|_F$, in [48]. For a fair comparison, the algorithm with theoretical guaranteed convergency in [53] is adopted to optimize these two models. The video "*foreman*" is selected with SRs varying from 10% to 50% for testing. We exhibit the PSNR and SSIM values of the results in Table V. These two traditional dictionary learning methods are, respectively, denoted as ℓ_1 and $\ell_{1,2}$. We can see from Table V that the $\ell_{1,2}$ constraint is also effective when the SR is bigger than 30% (see the SSIM values). However, when the SR is low, the margin between $\ell_{1,2}$ and our DTNN becomes much larger. Therefore, we can deduce that using the combination of dictionary's atoms (the low-rank constraint) would be helpful to eliminate the deviation caused by inaccurate estimation of the dictionary from incomplete data. This also supports our statement at the end of Section III-A that we need both the learned dictionary and the specific low-rank structure of the coefficients for accurate completion of the data.

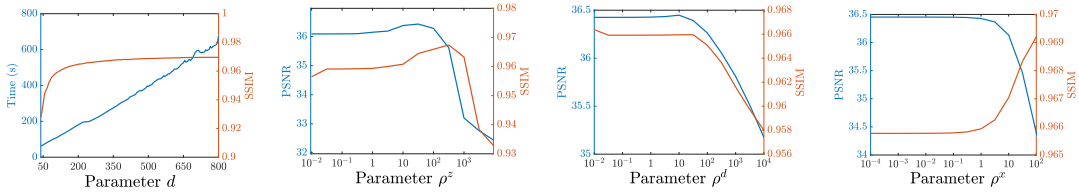


Fig. 6. Running time in seconds and SSIM values with different numbers of dictionary atoms (d), and PSNR and SSIM values of the result by our method with different ρ^z, ρ^d, and ρ^x, on the video "*foreman*" (SR = 50%).

TABLE V
PSNR AND SSIM VALUES OF RESULTS BY DTNN AND TRADITIONAL SPARSE REGULARIZERS WITH DIFFERENT SRS ON THE VIDEO DATA "*Foreman*"

SR	10%		20%		30%		40%		50%	
Method	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
ℓ_1	23.80	0.703	25.08	0.751	29.81	0.909	31.46	0.918	33.29	0.942
$\ell_{1,2}$	23.86	0.708	25.23	0.760	30.20	0.911	31.76	0.944	33.36	0.959
DTNN	**26.22**	**0.799**	**29.26**	**0.875**	**31.77**	**0.917**	**34.13**	**0.946**	**36.32**	**0.964**

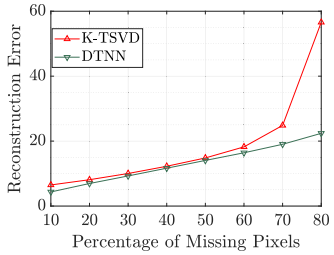


Fig. 7. RE values of results by K-TSVD and DTNN on the video "basketball" with respect to different percentages of missing pixels (from 10% to 80%).

2) Parameters: In our experiments, we find that four parameters mainly affect the performance of our method, i.e., the number of the dictionary atoms d and the proximal parameters ρ^z, ρ^d, and ρ^x. Although ρ^z, ρ^d, and ρ^x can be finely specified for each iteration, we respectively fix their values across our algorithm to reduce the parameter tuning burden. To test the effects from different values of them, we conduct experiments on the video "*foreman*" with setting the SR as 50%.

3) Comparisons With a Tensor Dictionary Learning Method: In this section, we compare our method with the tensor dictionary learning method (denoted as K-TSVD) in [48]. We implement our method and K-TSVD[15] on the video "basketball.[16]". The experimental setting for K-TSVD is the same as that in [48, Sec. 4.1]. That is, the first 30 frames are taken for training and the last ten frames for testing. We plot the reconstruction error (RE), which is defined as $\text{RE} = (\|\mathcal{X}_{\text{GT}} - \mathcal{X}_{\text{Rec}}\|_F^2/N)^{1/2}$ in [48], with respect to the percentage of missing pixels in Fig. 7. Lower RE values indicate better reconstructions. We can see that the results by our method are always better than those by K-TSVD. It is noteworthy that K-TSVD takes the first 30 frames as training data while our method does not have any extra training data and directly estimates the complete tensor from incomplete observations.

When testing one parameter, other three are fixed as default values. As for d, we vary its value from 40 ($0.8n_3$) to 800 ($16n_3$) with a step size 5. ρ^z and ρ^d are tested with candidates $\{10^{-2}, 10^{-1.5}, \ldots, 10^4\}$ while ρ^x varies from 10^4 to 10^2. We illustrate the running time and SSIM values with respect to different values of d and the PSNR and SSIM values with respect to different values of ρ^z, ρ^d, and ρ^x in Fig. 6. From Fig. 6, we can see that as d increases, the performance of our method becomes better while the running time also grows. Our default setting ($d = 5n_3 = 250$) is a compromise between effectiveness and efficiency. Meanwhile, we can see the performance of our method is more sensitive to ρ^z and our method could obtain satisfactory results with a wide range of ρ^d and ρ^x.

[15]The code for K-TSVD is modified from a public available implementation at https://github.com/takshingchan/ktsvd. The parameters of K-TSVD are manually selected for the best performance.

[16]https://sites.google.com/site/jamiezeminzhang/publications

TABLE VI
PSNR, SSIM, AND UIQI VALUES OF RESULTS BY OUR METHOD WITH DIFFERENT INITIALIZATIONS ON THE VIDEO DATA *Foreman* (SR = 50%). THE **Best** AND THE SECOND BEST VALUES ARE, RESPECTIVELY, HIGHLIGHTED BY **Boldface** AND UNDERLINE

	Method		PSNR	SSIM	UIQI
	Observed		6.51	0.047	0.056
	HaLRTC		31.85	0.928	0.853
	DCTNN		34.07	0.934	0.854
	$\mathcal{X}_0$	$\mathbf{D}_0$			
DTNN	Random	Random	14.65	0.175	0.197
		Tubes	13.29	0.133	0.153
	Interpolation*	Random	33.52	0.939	0.862
		Tubes*	36.32	0.964	0.907
	HaLRTC	Random	32.44	0.926	0.843
		Tubes	36.50	**0.965**	**0.908**
	DCTNN	Random	32.74	0.930	0.849
		Tubes	**36.51**	**0.965**	**0.908**

*Default setting in our experiments.

From Table VI, we can see that when $\mathcal{X}_0$ is randomly initialized, the performance of our method is poor. When implementing our method with $\mathcal{X}_0$s using the results from HaLRTC and DCTNN, the performances are better than default setting. This shows that our method indeed relies on initialization as many nonconvex optimization methods. As for $\mathbf{D}_0$, our method performs well when using tubes of $\mathcal{X}_0$, as it would contribute to flexibility of $\mathbf{D}$.

4) Initializations: In this section, we test different initialization strategies. Other than default setting, we use the random tensor, whose values are uniformly distributed in the interval [0, 1], and the results from HaLRTC and DCTNN, which are fast, as initial guesses of $\mathcal{X}_0$. Meanwhile, we also initialize the dictionary using random values following a standard normal distribution. Also, we take the video "*foreman*" with SR 50% as an example. The results are shown in Table VI.

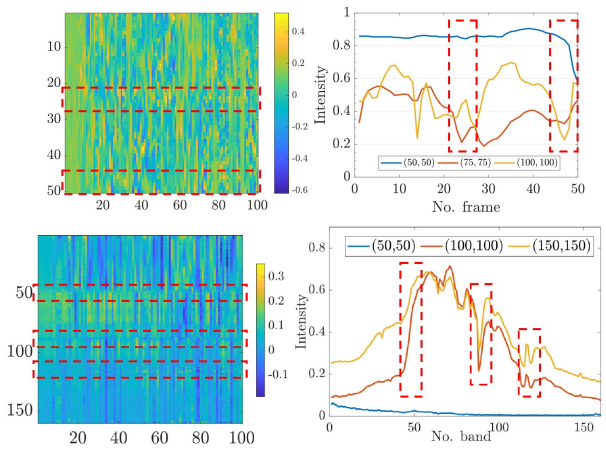


Fig. 8. Learned dictionaries (left) and the tubes of original data (right). Top: video "*foreman*" of size $144 \times 176 \times 50$ with SR $= 50\%$. Bottom: the HSI Washinton DC Mall of size $256 \times 256 \times 160$ with SR $= 40\%$.

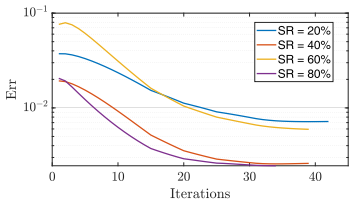


Fig. 9. Estimation error (Err) of the dictionary with respect to iterations for synthetic data.

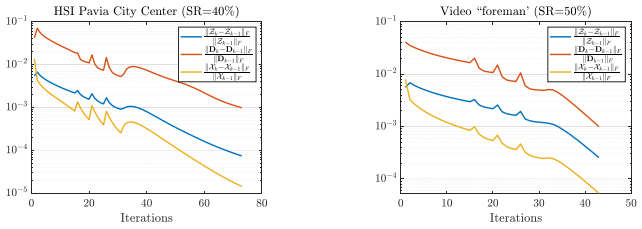


Fig. 10. Relative changes in the variables. Left: MRI data with SR $= 30\%$. Right: video data "foreman" with SR $= 50\%$.

5) Learned Dictionaries: In Fig. 8, we exhibit the first 100 columns of the learned dictionaries together with the plotting of three tubes of the original data. From the red boxes with dashed line, we can see that when the tubes, i.e., the vectors along the third dimension, of the original data fluctuate, the corresponding areas of the dictionaries' atoms (columns) tend not to be smooth. This reflects that the dictionaries learned by our method are flexible and adaptive to different types of data.

Meanwhile, we simulate a tensor $\mathcal{X} \in \mathbb{R}^{50\times50\times50} = \mathcal{Z} \times_3 \mathbf{D}$, where frontal slices of $\mathcal{Z} \in \mathbb{R}^{50\times50\times250}$ are obtained via multiplication between randomly generated matrices of sizes 50×5 and 5×50, and $\mathbf{D} \in \mathbb{R}^{50\times250}$ is randomly generated and normalized with the norm of its columns equaling to 1. Thus, we have the ground truth of the dictionary and we adopt the estimation error (Err),[17] of the dictionary [65] as a quantitative metric to measure the accuracy of the estimated dictionary. We plot the estimation errors with respect to iteration numbers in Fig. 9. Although the initial Err values are different owing to the initilization stage, we can see that Err is becoming smaller as iteration goes on. This shows that our method could enforce the estimated dictionary being close to the ground truth under different SRs.

[17]The estimation error is defined as $\text{Err} = (1/d)\sum_{i=1}^{d}(1 - |(\mathbf{d}^i_{\text{Est}})^\top \mathbf{d}^{i_0}_{\text{GT}}|)$ where $\mathbf{d}^i_{\text{Est}}$ is the ith atom (column) of the estimated dictionary, $\mathbf{d}^{i_0}_{\text{GT}}$ is the i_0th atom of the ground-truth dictionary, and $i_0 = \arg\max_{j\in\mathbb{N}^+, 1\le j\le d, j\neq i} |(\mathbf{d}^i_{\text{Est}})^\top \mathbf{d}^j_{\text{GT}}|$.

TABLE VII

COMPUTATIONAL COMPLEXITY OF EACH METHOD TO DEAL WITH A TENSOR WITH SIZE $n_1 \times n_2 \times n_3$, AND THE AVERAGED ITERATIONS NEEDED FOR DIFFERENT TYPES OF DATA

Method	Complexity per iteration	Iterations			
		Video	HSI	MRI	Traffic
HaLRTC	$\mathrm{O}(n_1n_2n_3\sum_{j=1}^{3} n_j)$	59	97	52	108
BCPF	$\mathrm{O}(3R^2\lvert\Omega\rvert + R^3)$	14	14	12	15
TRLRF	$\mathrm{O}(R^2n_1n_2n_3 + R^6)$	487	500	493	500
TNN	$\mathrm{O}(n_1n_2n_3(\log n_3 + \min(n_1, n_2))$	86	77	76	94
DCTNN	$\mathrm{O}(n_1n_2n_3(\log n_3 + n_1)$	97	91	88	103
FTNN	$\mathrm{O}(\omega n_1n_2n_3(n_3 + \min(n_1, n_2))$	86	79	40	68
DTNN	$\mathrm{O}(dn_1n_2(dn_3 + \min(n_1, n_2) + n_3)$	61	72	62	52

6) Convergency Behaviors: When the largest relative change in the variables, i.e., $\max\{(\|\mathcal{Z}_k - \mathcal{Z}_{k-1}\|_F)/(\|\mathcal{Z}_{k-1}\|_F), (\|\mathbf{D}_k - \mathbf{D}_{k-1}\|_F)/(\|\mathbf{D}_{k-1}\|_F), (\|\mathcal{X}_k - \mathcal{X}_{k-1}\|_F)/(\|\mathcal{X}_{k-1}\|_F)\}$, is smaller than 10^{-3}, we consider that our algorithm converges and stop the iterations. In Fig. 10, we present the relative changes with respect to the iterations in our experiments on the HSI data Pavia City Center and the video data "*foreman*." Three obvious fluctuations in each curve are in accord with our parameter setting of enlarging ρ at the 15th, 20th, and 25th iterations. The overall downward trend of the curves in Fig. 10 illustrates that our method converges quickly.

Moreover, we list the computational complexity of the compared methods and the iterations needed for different types of data in Table VII. The CP rank used in BCPF is R and the TR rank used in TRLRF is $[R, R, R]$. For FTNN, ω corresponds to the construction of the framelet system. Although our method generally needs fewer iterations than other TNN-induced methods (TNN, DCTNN, and FTNN), it costs more running time. The main reason is that the computation complexity of our method is high as d is much bigger than n_3.

V. CONCLUSION

In this article, we have introduced the data-adaptive dictionary and low-rank coding for third-order tensor completion. In the completion model, we have proposed to minimize the low-rankness of each tensor slice containing the coding coefficients. To optimize this model, we design a multiblock proximal alternating minimization algorithm, the sequence generated by which would globally converge to a critical point. Numerical experiments conducted on various types of real-world data show the superiority of the proposed method.

As a future research work, we will consider how to use the proposed model and idea for a wider range of applications, such as tensor robust principal component analysis [43], [66], tensor-based representation learning for multiview clustering [67], [68], and other challenging image/video restoration tasks [45], [69], [70].

ACKNOWLEDGMENT

The authors would like to thank the editor and reviewers for giving them many comments and suggestions, which are of great value for improving the quality of this manuscript. They would like to thank Liu *et al.* [4], Zhao *et al.* [14], Yuan *et al.* [24], Zhang and Aeron [33], [48], and Lu [71] for their generous sharing of their codes or data.

REFERENCES

[1] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proc. Annu. Conf. Comput. Graph. Interact. Techn. (SIGGRAPH)*, 2000, pp. 417–424.

[2] N. Komodakis, "Image completion using global optimization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2006, pp. 442–452.

[3] T. Korah and C. Rasmussen, "Spatiotemporal inpainting for recovering texture maps of occluded building facades," *IEEE Trans. Image Process.*, vol. 16, no. 9, pp. 2262–2271, Sep. 2007.

[4] 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.

[5] X.-L. Zhao, W.-H. Xu, T.-X. Jiang, Y. Wang, and M. K. Ng, "Deep plug-and-play prior for low-rank tensor completion," *Neurocomputing*, vol. 400, pp. 137–149, Dec. 2020.

[6] R. Koller *et al.*, "High spatio-temporal resolution video with compressed sensing," *Opt. Exp.*, vol. 23, no. 12, p. 15992, 2015.

[7] V. N. Varghees, M. S. Manikandan, and R. Gini, "Adaptive MRI image denoising using total-variation and local noise estimation," in *Proc. Int. Conf. Adv. Eng., Sci. Manage. (ICAESM)*, 2012, pp. 506–511.

[8] L. Zhuang and J. M. Bioucas-Dias, "Fast hyperspectral image denoising and inpainting based on low-rank and sparse representations," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 3, pp. 730–742, Mar. 2018.

[9] J. M. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. M. Nasrabadi, and J. Chanussot, "Hyperspectral remote sensing data analysis and future challenges," *IEEE Geosci. Remote Sens. Mag.*, vol. 1, no. 2, pp. 6–36, Jun. 2013.

[10] T.-X. Jiang, L. Zhuang, T.-Z. Huang, X.-L. Zhao, and J. M. Bioucas-Dias, "Adaptive hyperspectral mixed noise removal," *IEEE Trans. Geosci. Remote Sens.*, early access, Jun. 15, 2021, doi: 10.1109/TGRS.2021.3085779.

[11] R. Dian, S. Li, and X. Kang, "Regularizing hyperspectral and multispectral image fusion by CNN denoiser," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 3, pp. 1124–1135, Mar. 2021.

[12] A. H. Kiers, "Towards a standardized notation and terminology in multiway analysis," *J. Chemometrics*, vol. 14, no. 3, pp. 105–122, 2000.

[13] C. J. Hillar and L.-H. Lim, "Most tensor problems are NP-hard," *J. ACM*, vol. 60, no. 6, pp. 1–39, Nov. 2013.

[14] Q. Zhao, L. Zhang, and A. Cichocki, "Bayesian CP factorization of incomplete tensors with automatic rank determination," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1751–1763, Sep. 2015.

[15] Z. Han *et al.*, "A generalized model for robust tensor factorization with noise modeling by mixture of Gaussians," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 11, pp. 5380–5393, Nov. 2018.

[16] Q. Zhao, G. Zhou, L. Zhang, A. Cichocki, and S.-I. Amari, "Bayesian robust tensor factorization for incomplete multiway data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 4, pp. 736–748, Apr. 2016.

[17] L. R. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, no. 3, pp. 279–311, 1966.

[18] X. Zhang, "A nonconvex relaxation approach to low-rank tensor completion," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 6, pp. 1659–1671, Jun. 2019.

[19] I. V. Oseledets, "Tensor-train decomposition," *SIAM J. Sci. Comput.*, vol. 33, no. 5, pp. 2295–2317, 2011.

[20] J. A. Bengua, H. N. Phien, H. D. Tuan, and M. N. Do, "Efficient tensor completion for color image and video recovery: Low-rank tensor train," *IEEE Trans. Image Process.*, vol. 26, no. 5, pp. 2466–2479, May 2017.

[21] R. Dian, S. Li, and L. Fang, "Learning a low tensor-train rank representation for hyperspectral image super-resolution," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 9, pp. 2672–2683, Sep. 2019.

[22] Y. Liu, J. Liu, and C. Zhu, "Low-rank tensor train coefficient array estimation for tensor-on-tensor regression," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 12, pp. 5402–5411, Dec. 2020.

[23] Q. Zhao, G. Zhou, S. Xie, L. Zhang, and A. Cichocki, "Tensor ring decomposition," 2016, *arXiv:1606.05535*. [Online]. Available: http://arxiv.org/abs/1606.05535

[24] L. Yuan, C. Li, D. Mandic, J. Cao, and Q. Zhao, "Tensor ring decomposition with rank minimization on latent space: An efficient approach for tensor completion," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 9151–9158.

[25] J. Yu, G. Zhou, C. Li, Q. Zhao, and S. Xie, "Low tensor-ring rank completion by parallel matrix factorization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 7, pp. 3020–3033, Jul. 2021, doi: 10.1109/TNNLS.2020.3009210.

[26] Z. Long, Y. Liu, L. Chen, and C. Zhu, "Low rank tensor completion for multiway visual data," *Signal Process.*, vol. 155, pp. 301–316, Feb. 2019.

[27] Q. Song, H. Ge, J. Caverlee, and X. Hu, "Tensor completion algorithms in big data analytics," *ACM Trans. Knowl. Discovery Data*, vol. 13, no. 1, pp. 1–48, Jan. 2019.

[28] K. Braman, "Third-order tensors as linear operators on a space of matrices," *Linear Algebra Appl.*, vol. 433, no. 7, pp. 1241–1253, 2010.

[29] M. E. Kilmer and C. D. Martin, "Factorization strategies for third-order tensors," *Linear Algebra Appl.*, vol. 435, no. 3, pp. 641–658, 2011.

[30] N. Hao, M. E. Kilmer, K. Braman, and R. C. Hoover, "Facial recognition using tensor-tensor decompositions," *SIAM J. Imag. Sci.*, vol. 6, no. 1, pp. 437–463, 2013.

[31] M. E. Kilmer, K. Braman, N. Hao, and R. C. Hoover, "Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging," *SIAM J. Matrix Anal. Appl.*, vol. 34, no. 1, pp. 148–172, 2013.

[32] Z. Zhang, G. Ely, S. Aeron, N. Hao, and M. Kilmer, "Novel methods for multilinear data completion and de-noising based on tensor-SVD," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 3842–3849.

[33] Z. Zhang and S. Aeron, "Exact tensor completion using t-SVD," *IEEE Trans. Signal Process.*, vol. 65, no. 6, pp. 1511–1526, Mar. 2017.

[34] J. Q. Jiang and M. K. Ng, "Robust low-tubal-rank tensor completion via convex optimization," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 2649–2655.

[35] A. Wang, X. Song, X. Wu, Z. Lai, and Z. Jin, "Robust low-tubal-rank tensor completion," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 3432–3436.

[36] E. Kernfeld, M. Kilmer, and S. Aeron, "Tensor–tensor products with invertible linear transforms," *Linear Algebra Appl.*, vol. 485, pp. 545–570, Nov. 2015.

[37] 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 Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 5996–6004.

[38] W.-H. Xu, X.-L. Zhao, and M. Ng, "A fast algorithm for cosine transform based tensor singular value decomposition," 2019, *arXiv:1902.03070*. [Online]. Available: http://arxiv.org/abs/1902.03070

[39] 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, May 2020, Art. no. e2299.

[40] T.-X. Jiang, M. K. Ng, X.-L. Zhao, and T.-Z. Huang, "Framelet representation of tensor nuclear norm for third-order tensor completion," *IEEE Trans. Image Process.*, vol. 29, pp. 7233–7244, 2020.

[41] Y.-B. Zheng, T.-Z. Huang, X.-L. Zhao, T.-X. Jiang, T.-Y. Ji, and T.-H. Ma, "Tensor N-tubal rank and its convex relaxation for low-rank tensor recovery," *Inf. Sci.*, vol. 532, pp. 170–189, Sep. 2020.

[42] C. D. Martin, R. Shafer, and B. LaRue, "An order-p tensor factorization with applications in imaging," *SIAM J. Sci. Comput.*, vol. 35, no. 1, pp. A474–A490, 2013.

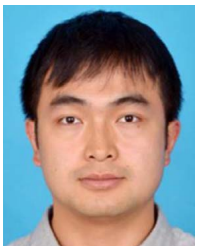
[43] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan, "Tensor robust principal component analysis with a new tensor nuclear norm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 4, pp. 925–938, Apr. 2020.

[44] Y.-Y. Liu, X.-L. Zhao, Y.-B. Zheng, T.-H. Ma, and H. Zhang, "Hyperspectral image restoration by tensor fibered rank constrained optimization and plug-and-play regularization," *IEEE Trans. Geosci. Remote Sens.*, early access, Jan. 5, 2021, doi: 10.1109/TGRS.2020.3045169.

[45] J.-H. Yang, X.-L. Zhao, T.-H. Ma, Y. Chen, T.-Z. Huang, and M. Ding, "Remote sensing images destriping using unidirectional hybrid total variation and nonconvex low-rank regularization," *J. Comput. Appl. Math.*, vol. 363, pp. 124–144, Jan. 2020.

[46] D. L. Donoho, "For most large underdetermined systems of linear equations the minimal ℓ_1-norm solution is also the sparsest solution," *Commun. Pure Appl. Math.*, vol. 59, no. 6, pp. 797–829, 2006.

[47] Y. Peng, D. Meng, Z. Xu, C. Gao, Y. Yang, and B. Zhang, "Decomposable nonlocal tensor dictionary learning for multispectral image denoising," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2949–2956.
[48] Z. Zhang and S. Aeron, "Denoising and completion of 3D data via multidimensional dictionary learning," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2016, pp. 2371–2377.
[49] D. Geman and G. Reynolds, "Constrained restoration and the recovery of discontinuities," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 3, pp. 367–383, Mar. 1992.
[50] M. Nikolova and M. K. Ng, "Analysis of half-quadratic minimization methods for signal and image recovery," *SIAM J. Sci. Comput.*, vol. 27, no. 3, pp. 937–966, 2005.
[51] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.
[52] J. Bolte, S. Sabach, and M. Teboulle, "Proximal alternating linearized minimization for nonconvex and nonsmooth problems," *Math. Program.*, vol. 146, no. 1, pp. 459–494, Aug. 2014.
[53] C. Bao, H. Ji, Y. Quan, and Z. Shen, "Dictionary learning for sparse coding: Algorithms and convergence analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 7, pp. 1356–1369, Jul. 2016.
[54] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.
[55] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. Optim.*, vol. 20, no. 4, pp. 1956–1982, 2010.
[56] H. Attouch, J. Bolte, and B. F. Svaiter, "Convergence of descent methods for semi-algebraic and tame problems: Proximal algorithms, forward-backward splitting, and regularized Gauss–Seidel methods," *Math. Program.*, vol. 137, no. 1, pp. 91–129, 2013.
[57] R. T. Rockafellar and R. J.-B. Wets, *Variational Analysis*, vol. 317. Berlin, Germany: Springer-Verlag, 2009.
[58] F. H. Clarke, Y. S. Ledyaev, R. J. Stern, and P. R. Wolenski, *Nonsmooth Analysis and Control Theory*, vol. 178. New York, NY, USA: Springer-Verlag, 2008.
[59] H. Attouch, J. Bolte, P. Redont, and A. Soubeyran, "Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka–Łojasiewicz inequality," *Math. Oper. Res.*, vol. 35, no. 2, pp. 438–457, May 2010.
[60] J. Bolte, A. Daniilidis, A. Lewis, and M. Shiota, "Clarke subgradients of stratifiable functions," *SIAM J. Optim.*, vol. 18, no. 2, pp. 556–572, 2007.
[61] N. Yair and T. Michaeli, "Multi-scale weighted nuclear norm image restoration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3165–3174.
[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] Z. Wang and A. C. Bovik, "A universal image quality index," *IEEE Signal Process. Lett.*, vol. 9, no. 3, pp. 81–84, Mar. 2002.
[64] R. H. Yuhas, J. W. Boardman, and A. F. Goetz, "Determination of semi-arid landscape endmembers and seasonal trends using convex geometry spectral unmixing techniques," in *Proc. JPL, Summaries 4th Annu. JPL Airborne Geosci. Workshop*, vol. 1, 1993, pp. 147–149.
[65] Q. Yu, W. Dai, Z. Cvetković, and J. Zhu, "Dictionary learning with BLOTLESS update," *IEEE Trans. Signal Process.*, vol. 68, pp. 1635–1645, 2020.
[66] F. Zhang, J. Wang, W. Wang, and C. Xu, "Low-tubal-rank plus sparse tensor recovery with prior subspace information," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Apr. 13, 2020, doi: 10.1109/TPAMI.2020.2986773.
[67] M. Cheng, L. Jing, and M. K. Ng, "Tensor-based low-dimensional representation learning for multi-view clustering," *IEEE Trans. Image Process.*, vol. 28, no. 5, pp. 2399–2414, May 2019.
[68] 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.
[69] Y.-T. Wang, X.-L. Zhao, T.-X. Jiang, L.-J. Deng, Y. Chang, and T.-Z. Huang, "Rain streaks removal for single image via kernel-guided convolutional neural network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 8, pp. 3664–3676, Aug. 2021.
[70] T.-X. Jiang, T.-Z. Huang, X.-L. Zhao, L.-J. Deng, and Y. Wan, "Fastderain: A novel video rain streak removal method using directional gradient priors," *IEEE Trans. Image Process.*, vol. 28, no. 4, pp. 2089–2102, Apr. 2019.
[71] C. Lu. (Jun. 2018). Tensor-tensor product toolbox. Carnegie Mellon University. [Online]. Available: https://github.com/canyilu/tproduct



Tai-Xiang Jiang received the Ph.D. degree in mathematics from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2019. He was a co-training Ph.D. student with the University of Lisbon, Lisbon, Portugal, from 2017 to 2018, supervised by Prof. Jose M. Bioucas-Dias.

He was a Research Assistant with the Hong Kong Baptist University, Hong Kong, supported by Prof. Michael K. Ng in 2019. He is currently an Associate Professor with the School of Economic Information Engineering, Southwestern University of Finance and Economics, Chengdu. His research interests include sparse and low-rank modeling and tensor decomposition for multidimensional image processing, especially on low-level inverse problems for multidimensional images. https://taixiangjiang.github.io/

Xi-Le Zhao received the M.S. and Ph.D. degrees from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2009 and 2012, respectively.

He is currently a Professor with the School of Mathematical Sciences, UESTC. His main research interest is focused on the models and algorithms of high-dimensional image processing problems.

Hao Zhang received the B.S. degree from the College of Science, Sichuan Agricultural University (SAU), Ya'an, China, in 2018. He is currently pursuing the M.S. degree with the School of Mathematical Sciences, University of Electronic Science and Technology of China (UESTC), Chengdu, China.

His research interest is modeling and algorithm for high-order data recovery based on the low-rank prior of tensors.

Michael K. Ng is the Director of the Research Division for Mathematical and Statistical Science, the Chair Professor of the Department of Mathematics, The University of Hong Kong, Hong Kong, and the Chairman of the HKU-TCL Joint Research Center for AI.

His research areas are data science, scientific computing, and numerical linear algebra.