Revisiting High-Order Tensor Singular Value Decomposition From Basic Element Perspective

Sheng Liu, Xi-Le Zhao, Jinsong Leng, Ben-Zheng Li, *Graduate Student Member, IEEE*, Jing-Hua Yang, *Member, IEEE*, and Xinyu Chen

***Abstract*—Recently, tensor singular value decomposition (t-SVD), based on the tensor-tensor product (t-product), has become a powerful tool for processing third-order tensor data. However, constrained by the fact that the basic element is the fiber (i.e., vector) in the t-product, higher-order tensor data (i.e., order $d > 3$) are usually unfolded into third-order tensors to satisfy the classical t-product setting, which leads to the destroying of high-dimensional structure. By revisiting the basic element in the t-product, we suggest a generalized t-product called element-based tensor-tensor product (elt-product) as an alternative of the classic t-product, where the basic element is a $(d-2)$th-order tensor instead of a vector. The benefit of the elt-product is that it can better preserve high-dimensional structures and that it can explore more complex interactions via higher-order convolution instead of first-order convolution in classic t-product. Starting from the elt-product, we develop new tensor-SVD and low-rank tensor metrics (e.g., rank and nuclear norm). Equipped with the suggested metrics, we present a tensor completion model for high-order tensor data and prove the exact recovery guarantees. To harness the resulting nonconvex optimization problem, we apply an alternating direction method of the multiplier (ADMM) algorithm with a theoretical convergence guarantee. Extensive experimental results on the simulated and real-world data (color videos, light-field images, light-field videos, and traffic data) demonstrate the superiority of the proposed model against the state-of-the-art baseline models.**

***Index Terms*—High-order tensor, low-rank tensor completion, element-based tensor-tensor product, tensor singular value decomposition, exact recovery guarantee.**

I. Introduction

TENSOR, as a generalization of matrices and vectors, provides a versatile and efficient framework for processing high-order tensor signals [1], [2], [3], [4], [5], [6], [7], such as hyperspectral images (HSIs) [8], [9], color videos [10], [11], light-field images [12], [13], and traffic data [14], [15], to name but a few. Nevertheless, observed signals in the real world often suffer from missing or corrupted conditions due to the limitations of imaging devices and environment, significantly hindering subsequent applications [16], [17]. Recovering underlying signals from the partially observed signals, i.e., tensor completion (TC), is a classical inverse problem in the signal processing field. Due to the inherent non-uniqueness and instability encountered in solving the inverse problem, it is an effective strategy to constrain the solution space by incorporating the prior information of signals. Low-rankness, as a prior, has been widely utilized for extracting the internal structure of signals in various real-world applications [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29].

To characterize the low-rank prior of high-order signals, tensor decompositions induced by different products of the tensor are investigated. The mainstream tensor decomposition includes CANDECOMP/PAEAFAC (CP) decomposition [30], Tukcer decomposition [31], [32], tensor train decomposition [33], [34], tensor ring decomposition [35], [36], [37], [38], and tensor singular value decomposition (t-SVD) [39], [40]. The goal of tensor decompositions is to decompose a given tensor into several mutually interacting factors, facilitating the exploration and extraction of low-rank structures. For example, the CP decomposition represents a tensor as the sum of the outer products of several vectors [30]. The Tucker decomposition is to decompose the tensor into a core tensor and factor matrices by using the tensor-matrix product [32]. Under the framework of the tensor network decomposition, the tensor is equivalent to the multiplication of several factors based on the tensor contraction [33], [35], [41]. Recently, t-SVD induced by tensor-tensor product (t-product) has gained widespread attention due

Received 29 November 2023; revised 22 June 2024 and 28 August 2024; accepted 29 August 2024. Date of publication 4 September 2024; date of current version 18 October 2024. This work was supported in part by NSFC under Grant 12371456, Grant 12171072, Grant 62131005, and Grant 12401605, in part by Sichuan Science and Technology Program under Grant 2024NSFJQ0038 and Grant 2023ZYD0007, in part by the National Key Research and Development Program of China under Grant 2020YFA0714001, in part by the Natural Science Foundation of Sichuan Province under Grant 2024NSFSC1467, in part by the Postdoctoral Fellowship Program of CPSF under Grant GZC20232198, in part by the Fundamental Research Funds for the Central Universities under Grant 2682024CX017, in part by the China Postdoctoral Science Foundation under Grant 2024M752661, and in part by the Fund of Hubei Key Laboratory of Inland Shipping Technology under Grant NHHY2023003. The associate editor coordinating the review of this article and approving it for publication was Dr. Tatsuya Yokota. *(Corresponding author: Xi-Le Zhao.)*

Sheng Liu, Xi-Le Zhao, Jinsong Leng, and Ben-Zheng Li are with the School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu 611731, China (email: liusheng16@163.com; xlzhao122003@163.com; lengjs@uestc.edu.cn; lbz1604179601@126.com).

Jing-Hua Yang is with the School of Information Science and Technology, Southwest Jiaotong University, Chengdu 611756, China (e-mail: yangjinghua110@126.com).

Xinyu Chen is with the Civil, Geological and Mining Engineering Department, Polytechnique Montreal, Montreal, QC H3T 1J4, Canada (e-mail: chenxy346@gmail.com).

This article has supplementary downloadable material available at https://doi.org/10.1109/TSP.2024.3454115, provided by the authors.

Digital Object Identifier 10.1109/TSP.2024.3454115

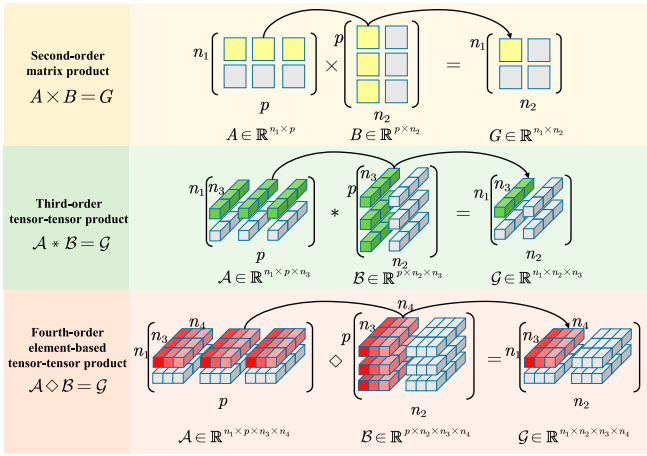


Fig. 1. Illustration of the representation forms of the matrix product, tensor-tensor product, and element-based tensor-tensor product. The basic element of the matrix product is the zero-order scalar (yellow part in Fig. 1); the basic element of the classical tensor-tensor product is the vector (green part in Fig. 1); the basic element of the element-based tensor-tensor product is the $(d-2)$th-order tensor, e.g., matrix (red part in Fig. 1).

to its remarkable ability to capture spatial-shifting correlations along the mode-3 direction in real signals [39], [42], [43], [44], [45], [46], [47]. Based on the t-SVD, a third-order tensor of size $n_1 \times n_2 \times n_3$ can be decomposed into two orthogonal tensors and an f-diagonal tensor via the t-product, where the basic element of the t-product is a fiber of size $1 \times 1 \times n_3$. Many excellent works focus on designing the better transform matrices to improve the performance of the classical t-SVD [39] for third-order tensors [48], [49], [50], [51], [52], [53], [54], [55], [56]. However, the emergence of higher-order tensor data (e.g., color videos, light-field images, and light-field videos) from real-world applications beckons for analyzing higher-order signal processing tools suited to them.

To this end, the basic element that served as the foundation for defining the product should be expanded. From Fig. 1, we can see that, in the definition of the t-product, the first-order vector is adopted as the basic element instead of the zero-order scalar, distinguishing it from the matrix product. Thanks to this strategy, the models based on the t-product can effectively capture the low-rank structure of signals on the third-order tensor, specifically along the mode-3 direction [39], [44], [45]. However, for higher-order tensor signals (i.e., order $d > 3$), to satisfy the classical t-product setting where the basic element is a vector, they are usually unfolded into third-order tensors [57], [58], which hinders the natural representation of the higher-order tensor data. To address this issue, we suggest adopting the $(d-2)$th-order tensor instead of the vector as the basic element for the dth-order tensor in the definition of the product. It has two advantages compared to the classical t-product. Firstly, the new basic element can avoid unfolding operations to better preserve high-dimensional structures. Secondly, the corresponding operation of the basic element can explore more complex interactions via higher-order convolution instead of first-order convolution. Furthermore, to strengthen the interrelation between the different modes, it is imperative to extend the product from one mode to any mode.

Building upon the analysis above, in this paper, we revisit the classic t-product from the basic element perspective and suggest a generalized t-product called element-based tensor-tensor product (elt-product), which extends the classical t-product to dth-order. We suggest that the basic element should be a $(d-2)$th-order tensor instead of limiting the vector, thereby preserving the intrinsic structure of the high-order tensor data. Further, we can naturally extend the elt-product from one mode to any mode. This strengthens the interrelation among various elt-products defined in different modes, effectively overcoming the permutation sensitivity issue in classical t-product. Based on the proposed elt-product, we can define the corresponding tensor-SVD, tensor rank, and tensor nuclear norm. The main contributions of this paper are summarized as three-fold:

1) By revisiting the basic element in the t-product, we suggest a generalized t-product (elt-product) as an alternative of the classic t-product, where the basic element is a $(d-2)$th-order tensor instead of a vector. The benefit of the elt-product is that it can better preserve high-dimensional structures and that it can explore more complex interactions via higher-order convolution instead of first-order convolution in classic t-product.
2) Starting from the elt-product, we develop new tensor-SVD and low-rank tensor metrics (e.g., rank and nuclear norm). Equipped with the suggested metrics, we establish the corresponding low-rank tensor completion (LRTC) model for high-order tensor data and prove the exact recovery guarantee of the proposed model.
3) To harness the resulting nonconvex optimization problem, we apply an alternating direction method of the multiplier (ADMM) algorithm and prove the convergence of the developed algorithm. Extensive experimental results on the simulated and real-world data demonstrate the superiority of the proposed models against the state-of-the-art baseline models.

The rest of the paper is organized as follows. In Section II, we review some related works on the generalization of t-SVD for high-order tensor data. In Section III, we introduce some notations and preliminary knowledge. In Section IV, we present an elt-product and give a description of the corresponding tensor-SVD, tensor rank, and tensor nuclear norm. In Section V, we propose the LRTC model, in the meanwhile proving the exact recovery guarantee of the model. The experimental results are illustrated and discussed in Section VI. The conclusion is drawn in Section VII.

II. Related Work

To deal with high-order tensor data, researchers have developed generalizations of the classical t-SVD [57], [58], [59], [60], [61]. These works can be broadly classified into two categories: unfolding-based methods and recursion-based methods.

- Zheng et al. [57] proposed the weighted sum of the tensor nuclear norm (WSTNN) by unfolding the high-order tensor data into the third-order tensor along mode-k_1k_2. Similarly, Wang et al. [58] used mode-(k,t) 3D unfolding to unfold the high-order tensor data into the third-order

TABLE I
NOTATIONS AND EXPLANATIONS

Notations	Explanations
$[d]$	The set of the first d natural numbers.
x, $\boldsymbol{x}$, $\boldsymbol{X}$, $\mathcal{X}$	Scalar, vector, matrix, tensor.
$\mathcal{X}_{i_1 i_2 \cdots i_d}$	The $i_1 i_2 \cdots i_d$-th element of $\mathcal{X}$.
$\boldsymbol{X}^\top$	The transposition of $\boldsymbol{X} \in \mathbb{R}^{n_1 \times n_2}$.
$\sigma_i(\boldsymbol{X})$	The i-th singular value of $\boldsymbol{X}$.
$\|\boldsymbol{X}\|_* = \sum \sigma_i(\boldsymbol{X})$	The nuclear norm of $\boldsymbol{X}$.
$\langle \mathcal{X}, \mathcal{Y} \rangle$	The inner product between $\mathcal{X}$ and $\mathcal{Y}$.
$\|\mathcal{X}\|_F$	The Frobenius norm of $\mathcal{X}$.
$\mathrm{Tr}(\boldsymbol{X})$	The trace of $\boldsymbol{X}$.
$\boldsymbol{X}^{(i)}$	The i-th frontal slice of $\mathcal{X}$.
$\boldsymbol{X}_{(k)} = \mathrm{unfold}_k(\mathcal{X})$	The mode-k matricization of $\mathcal{X}$.
$\mathcal{X} = \mathrm{fold}_k(\boldsymbol{X}_{(k)})$	The inverse operation of $\mathrm{unfold}_k(\mathcal{X})$.

tensor to describe the low-rank structure. Both these methods unfold the dth-order tensor into a third-order tensor and then apply the t-SVD of the third-order to the unfolded tensor, which destroys the intrinsic structure of the high-order tensor data.

- Martin et al. [59] took a recursive manner, treating products of dth-order as successive t-product operations with $(d-1)$th-order tensors, and finally recursing to t-product of third-order. In addition, Qin et al. [60] suggested a strategy that performed a one-dimensional transform from the third dimension to the dth dimension for the tube of each dimension. Essentially, these two methods are equivalent as they concentrate on the third dimension to the dth dimension while disregarding the first and second dimensions, indicating the model's direction sensitivity. Beyond that, Wang et al. [61] proposed that higher order tensor SVD utilizes tubal-tensor and tubal-matrix to replace the core tensor and factor matrix in higher order SVD, where the tubal-tensor can be regarded as an $(d-1)$th-order tensor with the tubal as the basic element.

In this paper, the proposed elt-product can avoid the unfolding operation, it is extremely competitive compared to the classical t-product in exploring the intrinsic structure of high-order tensor data. In the meantime, we can naturally extend the elt-product from one mode to any mode, making it invariant to permutation, which overcomes the shortcomings of directional sensitivity.

III. NOTATIONS AND PRELIMINARIES

Throughout this paper, the notations are listed in Table I, and the preliminary knowledge used is summarized as follows:

Definition 1 (Orthogonal matrix, semi-orthogonal matrix): For a square matrix $\boldsymbol{X} \in \mathbb{R}^{n \times n}$, $\boldsymbol{X}$ is an orthogonal matrix if and only if it satisfies $\boldsymbol{X}\boldsymbol{X}^\top = \boldsymbol{I}$ and $\boldsymbol{X}^\top \boldsymbol{X} = \boldsymbol{I}$, where $\boldsymbol{I}$ is the identity matrix. For any matrix $\boldsymbol{X} \in \mathbb{R}^{m \times n}$, $\boldsymbol{X}$ is a semi-orthogonal matrix with orthogonal rows if and only if it satisfies $\boldsymbol{X}\boldsymbol{X}^\top = \boldsymbol{I}$. In addition, $\boldsymbol{X}$ is a semi-orthogonal matrix with orthogonal columns if and only if it satisfies $\boldsymbol{X}^\top \boldsymbol{X} = \boldsymbol{I}$.

Let $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \cdots \times n_d}$ be a dth-order tensor, and $\mathcal{X}_{\langle k \rangle} \triangleq \mathscr{P}_{\langle k \rangle}(\mathcal{X}) = \mathtt{permute}(\mathcal{X}, [k, k+1, \cdots, k-1+d]) \in \mathbb{R}^{n_k \times n_{k+1} \times \cdots \times n_{k-1+d}}$ for $\forall k \in [d]$ and $k-1+d := k-1$ with $k-1+d > d$. The corresponding inverse operation as $\mathscr{P}^{-1}_{\langle k \rangle}(\mathcal{X}) = \mathtt{ipermute}(\mathcal{X}, [k, k+1, \cdots, k-1+d])$, i.e., $\mathcal{X} = \mathscr{P}^{-1}_{\langle k \rangle}(\mathcal{X}_{\langle k \rangle})$.

For dth-order tensor $\mathcal{X}$, the mode-k high-dimensional block diagonal operation maps a dth-order tensor of size $n_1 \times n_2 \times \cdots \times n_d$ into a block diagonal matrix of size $n_k N_k \times n_{k+1} N_k$ with $N_k = n_1 n_2 \cdots n_{k-1} n_{k+2} \cdots n_d$, which is defined as follows

$$\mathrm{Hbdiag}(\mathcal{X}, k) = \begin{bmatrix} \boldsymbol{X}_k^{(1)} & & & & & \\ & \boldsymbol{X}_k^{(2)} & & & & \\ & & \ddots & & & \\ & & & \boldsymbol{X}_k^{(j_k)} & & \\ & & & & \ddots & \\ & & & & & \boldsymbol{X}_k^{(N_k)} \end{bmatrix},$$

where $\boldsymbol{X}_k^{(j_k)} = \mathcal{X}(i_1, i_2, \cdots, i_{k-1}, :, :, i_{k+2}, \cdots, i_d) \in \mathbb{R}^{n_k \times n_{k+1}}$, $j_k = i_{k+2} + !(i_{k+3} - 1) n_{k+2} + \cdots + (i_d - 1) n_{k+2} \cdots n_{d-1} + (i_1 - 1) n_{k+2} \cdots n_d + \cdots + (i_{k-1}) n_{k+2} \cdots n_d n_1 \cdots n_{k-2}, i_1 \in [n_1], i_2 \in [n_2], \cdots, i_{k-1} \in [n_{k-1}], i_{k+2} \in [n_{k+2}], \cdots, i_d \in [n_d]$.

Definition 2 (Mode-k tensor-matrix product [32]): For any dth-order tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times \cdots \times n_k \times \cdots \times n_d}$ with a matrix $\boldsymbol{U} \in \mathbb{R}^{n \times n_k}$, the mode-$k$ tensor-matrix product is denoted by $\mathcal{X} \times_k \boldsymbol{U} \in \mathbb{R}^{n_1 \times \cdots \times n \times \cdots \times n_d}$. For any element-wise, we have

$$(\mathcal{X} \times_k \boldsymbol{U})_{i_1 \cdots i_{k-1} j i_{k+1} \cdots i_d} = \sum_{i_k=1}^{n_k} x_{i_1 i_2 \cdots i_d} u_{j i_k}. \tag{1}$$

Based on the unfolding operator, the mode-k tensor-matrix product is equivalent to

$$\mathcal{Y} = \mathcal{X} \times_k \boldsymbol{U} \Leftrightarrow \boldsymbol{Y}_{(k)} = \boldsymbol{U}\boldsymbol{X}_{(k)}. \tag{2}$$

Definition 3 (Tensor-tensor product [39]): For third-order tensors $\mathcal{G} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, $\mathcal{A} \in \mathbb{R}^{n_1 \times p \times n_3}$, and $\mathcal{B} \in \mathbb{R}^{p \times n_2 \times n_3}$, the tensor-tensor product (t-product) is defined as

$$\mathcal{G} = \mathcal{A} * \mathcal{B} \Leftrightarrow \mathcal{G}(i, j, :) = \sum_{t=1}^{p} \mathcal{A}(i, t, :) \circledast \mathcal{B}(t, j, :), \tag{3}$$

where $\circledast$ denotes the circular convolution operation of two fibers. More specifically, the element-wise form of the circular convolution operation between $\boldsymbol{x} \in \mathbb{R}^{n_3}$ and $\boldsymbol{y} \in \mathbb{R}^{n_3}$ can be expressed as $(\boldsymbol{x} \circledast \boldsymbol{y})_k = \sum_{m=1}^{n_3} \boldsymbol{x}_{(k-m+1)} \boldsymbol{y}_m$, where $\boldsymbol{x}_{(k-m+1)} = \boldsymbol{x}_{(k-m+1+n_3)}$ if $k - m + 1 \le 0$, $k \in [n_3]$.

Definition 4 (Mode-k face-wise product): For any dth-order tensors $\mathcal{X} \in \mathbb{R}^{n_1 \times \cdots \times n_k \times p \times \cdots n_d}$ and $\mathcal{Y} \in \mathbb{R}^{n_1 \times \cdots \times p \times n_{k+1} \times \cdots n_d}$, the mode-$k$ face-wise product is denoted as $\mathcal{X} \triangle_k \mathcal{Y} \in \mathbb{R}^{n_1 \times \cdots n_k \times n_{k+1} \times \cdots \times n_d}$. For any element-wise, we have

$$(\mathcal{X} \triangle_k \mathcal{Y})_{i_1 \cdots i_k i_{k+1} \cdots i_d} = \sum_{t=1}^{p} x_{i_1 \cdots i_k t i_{k+2} \cdots i_d} y_{i_1 \cdots i_{k-1} t i_{k+1} \cdots i_d}. \tag{4}$$

Based on Hbdiag operator, the mode-k face-wise product is equivalent to

$$\begin{aligned}\mathcal{G} = \mathcal{X} \triangle_k \mathcal{Y} &\Leftrightarrow \mathrm{Hbdiag}(\mathcal{G}, k) \\ &= \mathrm{Hbdiag}(\mathcal{X}, k) \cdot \mathrm{Hbdiag}(\mathcal{Y}, k). \end{aligned} \tag{5}$$

Definition 4 is a generalization of the face-wise product from third-order to higher-order, where the third-order face-wise product is defined for three-order tensors by Kernfeld et al. in [49].

IV. High-Order Element-Based Tensor-Tensor Product

In this section, we present an elt-product and give new definitions of the corresponding tensor-SVD, tensor rank, and tensor nuclear norm. We begin with the preliminary definition of elt-product on the dth-order tensors.

Definition 5 (Element-based tensor-tensor product): For any dth-order tensors $\mathcal{A} \in \mathbb{R}^{n_1 \times p \times n_3 \times \cdots \times n_d}$, $\mathcal{B} \in \mathbb{R}^{p \times n_2 \times n_3 \times \cdots \times n_d}$, the element-based tensor-tensor product (elt-product) between these two tensors produces the dth-order tensor such that

$$\mathcal{G} = \mathcal{A} \lozenge \mathcal{B} \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times \cdots \times n_d}, \tag{6}$$

where $\lozenge$ denotes the operator of the tensor product.

More specifically, suppose $\boldsymbol{T}_1 \in \mathbb{R}^{n_1 \times n_1}$ and $\boldsymbol{T}_2 \in \mathbb{R}^{n_2 \times n_2}$ be the invertible transform matrices, then on each $\mathcal{G}^{(i,j)} \in \mathbb{R}^{n_3 \times n_4 \times \cdots \times n_d}, \forall i \in [n_1], j \in [n_2]$ (i.e., $(d-2)$th-order tensor), we have

$$\hat{\mathcal{G}}^{(i,j)} = \sum_{t=1}^{p} \hat{\mathcal{A}}^{(i,t)} \circledast \hat{\mathcal{B}}^{(t,j)}, \tag{7}$$

where the above variables are given by

$$\begin{cases} \hat{\mathcal{G}}^{(i,j)} \triangleq \mathcal{G} \times_1 \boldsymbol{T}_1(i,:) \times_2 \boldsymbol{T}_2(j,:), \\ \hat{\mathcal{A}}^{(i,t)} \triangleq \mathcal{A} \times_1 \boldsymbol{T}_1(i,:) \times_2 \boldsymbol{I}_p(t,:), \\ \hat{\mathcal{B}}^{(t,j)} \triangleq \mathcal{B} \times_1 \boldsymbol{I}_p(t,:) \times_2 \boldsymbol{T}_2(j,:), \end{cases} \tag{8}$$

in which $\boldsymbol{I}_p \in \mathbb{R}^{p \times p}$ is the identity matrix. The symbol $\circledast$ denotes the convolution with given boundary conditions[1] between two $(d-2)$th-order tensors, and $\times_1$ (or $\times_2$) denotes the mode-1 (or mode-2) product between tensor and matrix/vector. Specifically, for the convolution operation with periodic boundary conditions, the the element-wise form of the convolution operation between two $(d-2)$th-order tensors $\mathcal{X} \in \mathbb{R}^{n_3 \times n_4 \cdots \times n_N}$ and $\mathcal{Y} \in \mathbb{R}^{n_3 \times n_4 \cdots \times n_N}$ can be expressed as $(\mathcal{X} \circledast \mathcal{Y})_{k_3 \cdots k_N} = \sum_{m_3=1}^{n_3} \cdots \sum_{m_N=1}^{n_N} \mathcal{X}_{(k_3 - m_3 + 1) \cdots (k_N - m_N + 1)} \mathcal{Y}_{m_3 \cdots m_N}$, where $\mathcal{X}_{i_3 \cdots (k_j - m_j + 1) \cdots i_N} = \mathcal{X}_{i_3 \cdots (k_j - m_j + 1 + n_j) \cdots i_N}$ if $k_j - m_j + 1 \leq 0$, $\forall j \in \{3, 4, \cdots N\}$. In this special case, the corresponding transform matrix $\boldsymbol{T}_k$ is discrete Fourier matrix.

Definition 5 explains the correlation of $(d-2)$th-order tensor, while the correlation between two other modes (i.e., referring to two other orders) is totally ignored. This oversight may rise to direction-sensitive shortcomings in the elt-product possibly rendering the elt-product susceptible to direction-sensitive issues.

Definition 6 (Mode-k element-based tensor-tensor product): For any dth-order tensors $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots n_k \times p \times \cdots n_d}$, $\mathcal{B} \in \mathbb{R}^{n_1 \times \cdots p \times n_{k+1} \times \cdots n_d}$, and $\mathcal{G} \in \mathbb{R}^{n_1 \times \cdots n_k \times n_{k+1} \times \cdots n_d}$ with any invertible transform matrix $\boldsymbol{T}_k \in \mathbb{R}^{n_k \times n_k}$ and $\boldsymbol{T}_{k+1} \in \mathbb{R}^{n_{k+1} \times n_{k+1}}$, then the mode-$k$ element-based tensor-tensor product (elt-product), which is denoted by $\lozenge_k$, is defined as

$$\mathcal{G} = \mathcal{A} \lozenge_k \mathcal{B} \quad \Leftrightarrow \quad \hat{\mathcal{G}}^{(i,j)} = \sum_{t=1}^{p} \hat{\mathcal{A}}^{(i,t)} \circledast_k \hat{\mathcal{B}}^{(t,j)}, \tag{9}$$

where $\circledast_k$ denotes the convolution with given boundary conditions between $(d-2)$th-order tensors of size $n_1 \times \cdots \times n_{k-1} \times n_{k+2} \times \cdots \times n_d$. The above variables are given by

$$\begin{cases} \hat{\mathcal{G}}^{(i,j)} \triangleq \mathcal{G} \times_k \boldsymbol{T}_k(i,:) \times_{k+1} \boldsymbol{T}_{k+1}(j,:), \\ \hat{\mathcal{A}}^{(i,t)} \triangleq \mathcal{A} \times_k \boldsymbol{T}_k(i,:) \times_{k+1} \boldsymbol{I}_p(t,:), \\ \hat{\mathcal{B}}^{(t,j)} \triangleq \mathcal{B} \times_k \boldsymbol{I}_p(t,:) \times_{k+1} \boldsymbol{T}_{k+1}(j,:), \end{cases} \tag{10}$$

where $\times_k$ denotes the mode-k product.

Remark 1: The proposed product provides a flexible definition and includes some special cases as the existing product. As illustrated in Fig. 1, the matrix product, t-product, and elt-product work in the same way for constructing a matrix structure. In the special case that $k = 1$ and the invertible transform matrices $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$ are identity matrices, the elt-product is reduced to the classical t-product (Definition 3 [39]) on the third-order tensor. If $\mathcal{X}$ is a dth-order tensor, suppose $k = 1$ and the invertible transform matrices $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$ are identity matrices, then the elt-product is reduced to high-order t-product [59], [60]. Moreover, if the invertible transform matrices $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$ are identity matrices, then the elt-product refers to the mode-k t-product [62] on the third-order tensor.

To build the SVD of the elt-product, we need to introduce the fully transform and some special high-order tensors.

Definition 7 (Fully transform): For a dth-order tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ and any invertible transform matrix $\boldsymbol{T}_k \in \mathbb{R}^{n_k \times n_k}$, where $k \in [d]$. Then the full transform of $\mathcal{X}$ is defined as

$$\Phi(\mathcal{X}) \triangleq \mathcal{X} \times_1 \boldsymbol{T}_1^\top \times_2 \boldsymbol{T}_2^\top \times \cdots \times_d \boldsymbol{T}_d^\top. \tag{11}$$

The corresponding inverse operator is defined as

$$\Phi^{-1}(\mathcal{X}) \triangleq \mathcal{X} \times_1 \left(\boldsymbol{T}_1^\top\right)^{-1} \times_2 \left(\boldsymbol{T}_2^\top\right)^{-1} \times \cdots \times_d \left(\boldsymbol{T}_d^\top\right)^{-1}. \tag{12}$$

We also denote that $\Phi(\mathcal{X}) = \mathcal{X}_\Phi$ and $\Phi^{-1}(\mathcal{X}) = \mathcal{X}_{\Phi^{-1}}$.

Definition 8 (Mode-1 conjugate transpose tensor [59], [60]): The mode-1 conjugate transpose of a tensor $\mathcal{X}_1 \in \mathbb{C}^{n_1 \times n_2 \times n_3 \cdots \times n_d}$ is the tensor $\mathcal{X}_1^{\mathrm{H}_1} \in \mathbb{C}^{n_2 \times n_1 \times n_3 \cdots \times n_d}$ if the tensor slices always hold that

$$\mathcal{X}_{1,\Phi}^{\mathrm{H}_1}(:,:,i_3,\cdots,i_d) = (\mathcal{X}_{1,\Phi}(:,:,i_3,\cdots,i_d))^{\mathrm{H}_1}. \tag{13}$$

Definition 9 (Mode-1 identity tensor [59], [60]): The mode-1 identity tensor $\mathcal{I}_1 \in \mathbb{R}^{p \times p \times n_3 \times \cdots \times n_d}$ is the tensor with slices such that $\mathcal{I}_{1,\Phi}(:,:,i_3,\cdots,i_d) = \boldsymbol{I}_p$.

[1] More discussion about boundary conditions can be found in the supplementary material.

Definition 10 (Mode-1 f-diagonal tensor [59], [60]): The mode-1 f-diagonal tensor $\mathcal{X}_1 \in \mathbb{R}^{n\times n\times n_3\times\cdots n_d}$ is the tensor if and only if the tensor slice $\mathcal{X}_{1,\Phi}(:,:,i_3,\cdots,i_d)$ is a diagonal matrix.

Lemma 1: Given any dth-order tensors $\mathcal{G}\in\mathbb{R}^{n_1\times\cdots n_k\times n_{k+1}\times\cdots n_d}$, $\mathcal{X}\in\mathbb{R}^{n_1\times\cdots n_k\times p\times\cdots n_d}$, and $\mathcal{Y}\in\mathbb{R}^{n_1\times\cdots p\times n_{k+1}\times\cdots n_d}$, the following properties are always held:

1) $\mathrm{Hbdiag}(\mathcal{X},k)=\mathrm{Hbdiag}(\mathcal{X}_{\langle k\rangle},1)$.
2) $(\mathcal{X}^{\mathtt{H}_k})_{\langle k\rangle}=(\mathcal{X}_{\langle k\rangle})^{\mathtt{H}_1}$.
3) $\mathcal{G}_k=\mathcal{X}_k\Diamond_k\mathcal{Y}_k$ if and only if $\mathcal{G}_{\langle k\rangle}=\mathcal{X}_{\langle k\rangle}\Diamond_1\mathcal{Y}_{\langle k\rangle}$.
4) $\mathcal{X}$ is the mode-k identity tensor if and only if $\mathcal{X}_{\langle k\rangle}$ is the mode-1 identity tensor.
5) $\mathcal{X}$ is the mode-k diagonal tensor if and only if $\mathcal{X}_{\langle k\rangle}$ is the mode-1 diagonal tensor.
6) $\mathcal{X}$ is the mode-k orthogonal tensor if and only if $\mathcal{X}_{\langle k\rangle}$ is the mode-1 orthogonal tensor.

The proof of Lemma 1 can be found in the supplementary material.

Remark 2: Lemma 1 gives the connection between Definition 5 and Definition 6, demonstrating the importance of the permute operation for reinforcing Definition 5 (i.e., the direction of tensor product).

Definition 11 (Mode-k element-based tensor singular value decomposition): For any dth-order tensor $\mathcal{X}\in\mathbb{R}^{n_1\times n_2\times\cdots\times n_d}$, mode-$k$ element-based tensor singular value decomposition (elt-SVD) takes the factorization such that

$$\mathcal{X}=\mathcal{U}_k\Diamond_k\mathcal{S}_k\Diamond_k\mathcal{V}_k^{\mathtt{H}_k}, \tag{14}$$

where $\mathcal{U}_k\in\mathbb{R}^{n_1\cdots\times n_{k-1}\times n_k\times n_k\times n_{k+2}\times\cdots n_d}$ and $\mathcal{V}_k\in\mathbb{R}^{n_1\cdots\times n_{k-1}\times n_{k+1}\times n_{k+1}\times n_{k+2}\times\cdots n_d}$ are the mode-k orthogonal tensors and $\mathcal{S}_k\in\mathbb{R}^{n_1\cdots\times n_k\times n_{k+1}\times n_{k+2}\times\cdots n_d}$ is the mode-k diagonal tensor.

Definition 12 (Tensor-element multi-rank): The tensor-element multi-rank of $\mathcal{X}\in\mathbb{R}^{n_1\times\cdots\times n_d}$, denoted by $\mathrm{rank}_{tm}=R\in\mathbb{R}^{d\times M}$, is a matrix with its (k,i)-th entry being the rank of the i-th frontal slice of $\mathcal{X}_{\langle k\rangle,\Phi}$, i.e.,

$$R\triangleq\begin{bmatrix}\mathrm{rank}(\mathcal{X}^{(1)}_{\langle 1\rangle,\Phi}) & \mathrm{rank}(\mathcal{X}^{(2)}_{\langle 1\rangle,\Phi}) & \cdots & \mathrm{rank}(\mathcal{X}^{(\prod_{i\neq 1,2}^{d}n_i)}_{\langle 1\rangle,\Phi})\\ \mathrm{rank}(\mathcal{X}^{(1)}_{\langle 2\rangle,\Phi}) & \mathrm{rank}(\mathcal{X}^{(2)}_{\langle 2\rangle,\Phi}) & \cdots & \mathrm{rank}(\mathcal{X}^{(\prod_{i\neq 2,3}^{d}n_i)}_{\langle 2\rangle,\Phi})\\ \vdots & \vdots & \vdots & \vdots\\ \mathrm{rank}(\mathcal{X}^{(1)}_{\langle d\rangle,\Phi}) & \mathrm{rank}(\mathcal{X}^{(1)}_{\langle d\rangle,\Phi}) & \cdots & \mathrm{rank}(\mathcal{X}^{(\prod_{i\neq d,1}^{d}n_i)}_{\langle d\rangle,\Phi})\end{bmatrix}, \tag{15}$$

where $\mathrm{rank}(\mathcal{X}^{(\prod_{i\neq k,k+1}^{d}n_i)}_{\langle k\rangle,\Phi})=0$ if $\prod_{i\neq k,k+1}^{d}n_i<M$, $M=\max\{\prod_{i\neq k,k+1}^{d}n_i\},k\in[d]$, i.e., $R(k,t)=0$ when $\prod_{i\neq k,k+1}^{d}n_i<M$ and $\prod_{i\neq k,k+1}^{d}n_i\leq t\leq M$, and $\mathcal{X}_{\langle k\rangle,\Phi}=\Phi(\mathcal{X}_{\langle k\rangle})$.

Definition 13 (Mode-k tensor-element rank and tensor-element rank): The mode-k tensor-element rank of $\mathcal{X}\in\mathbb{R}^{n_1\times\cdots\times n_d}$ is defined as

$$\begin{aligned}\mathrm{rank}_{et,k}(\mathcal{X})&\triangleq\mathrm{rank}\left(\mathcal{X}_{\langle k\rangle}\right)=\max_t R(k,t)\\&=\#\{i:\mathcal{S}_{\langle k\rangle}(i,i,:,\cdots,:)\neq\mathbf{0}\}.\end{aligned} \tag{16}$$

The tensor-element rank of $\mathcal{X}$, denoted by $\mathrm{rank}_{et}(\mathcal{X})\in\mathbb{R}^d$, is a vector with its k-th entry being $\mathrm{rank}\left(\mathcal{X}_{\langle k\rangle}\right)$.

Since the definition of elt-product is associated with convolution operation, it may come at a high computational cost. To achieve fast and efficient computation, we also expect the proposed elt-product to keep some equivalence properties of the classical t-product in the transform domain. To this end, we give the following theorem.

Theorem 1: For any dth-order tensors $\mathcal{A}\in\mathbb{R}^{n_1\times\cdots n_k\times p\times\cdots n_d}$ and $\mathcal{B}\in\mathbb{R}^{n_1\times\cdots p\times n_{k+1}\times\cdots n_d}$ with any invertible transform $\boldsymbol{T}_k\in\mathbb{R}^{n_k\times n_k}$, the mode-$k$ elt-product (see Definition 6) is equivalent to

$$\mathcal{A}\Diamond_k\mathcal{B}=\Phi^{-1}(\Phi_{L_k}(\mathcal{A})\triangle_k\Phi_{R_k}(\mathcal{B})), \tag{17}$$

where

$$\Phi_{L_k}(\mathcal{X})\triangleq\mathcal{X}\times_1\boldsymbol{T}_1^\top\cdots\times_k\boldsymbol{T}_k^\top\times_{k+2}\boldsymbol{T}_{k+2}^\top\cdots\times_d\boldsymbol{T}_d^\top, \tag{18}$$

and the fully mode-k right transform of $\mathcal{X}$ is defined as

$$\Phi_{R_k}(\mathcal{X})\triangleq\mathcal{X}\times_1\boldsymbol{T}_1^\top\cdots\times_{k-1}\boldsymbol{T}_{k-1}^\top\times_{k+1}\boldsymbol{T}_{k+1}^\top\cdots\times_d\boldsymbol{T}_d^\top. \tag{19}$$

We also denote that $\Phi_{L_k}(\mathcal{X})=\mathcal{X}_{\Phi_{L_k}}$ and $\Phi_{R_k}(\mathcal{X})=\mathcal{X}_{\Phi_{R_k}}$.

The proof of Theorem 1 can be found in the supplementary material.

The aforementioned Theorem 1 allows one to build up the tensor nuclear norm induced by the elt-product.

Definition 14 (Mode-k element-based tensor nuclear norm): For $\mathcal{X}\in\mathbb{R}^{n_1\times n_2\times\cdots\times n_d}$ with any full column rank matrix $\boldsymbol{T}_k$, the mode-k element-based tensor nuclear norm (ETNN) is defined as

$$\begin{aligned}\|\mathcal{X}\|_{\mathrm{ETNN}_k}&\triangleq\|\mathrm{Hbdiag}(\mathcal{X}_\Phi,k)\|_*=\|\mathrm{Hbdiag}(\mathcal{C}_k,k)\|_*\\&=\sum_{i_1=1}^{n_1}\cdots\sum_{i_{k-1}=1}^{n_{k-1}}\sum_{i_{k+2}=1}^{n_{k+2}}\cdots\sum_{i_d=1}^{n_d}\|\mathcal{C}^{(i_1,\cdots,i_{k-1},:,:,i_{k+2},\cdots,i_d)}\|_*,\end{aligned} \tag{20}$$

where $\mathcal{X}=\mathcal{C}\times_1\boldsymbol{T}_1\times\cdots\times_d\boldsymbol{T}_d$ and $\mathcal{C}^{(i_1,\cdots,i_{k-1},i_{k+2},\cdots,i_d)}=\mathcal{C}(i_1,\cdots,i_{k-1},:,:,i_{k+2},\cdots,i_d)\in\mathbb{R}^{n_k\times n_{k+1}}$.

Remark 3: We extend the transform matrix $\boldsymbol{T}_k$ from invertible to non-invertible in Definition 14, which brings to better performance.

We define the enhanced ETNN as $\|\mathcal{X}\|_{\mathrm{ETNN}_k^+}$ when $\boldsymbol{T}_k$ is a non-invertible transform matrix. In this paper, we focus on orthogonal and semi-orthogonal transform matrices as representatives of invertible and non-invertible transform matrices.

Definition 15 (Multi-mode element-based tensor nuclear norm): For $\mathcal{X}\in\mathbb{R}^{n_1\times n_2\times\cdots\times n_d}$ with any orthogonal transform $\boldsymbol{T}_k\in\mathbb{R}^{n_k\times n_k}$, the multi-mode element-based tensor nuclear norm with orthogonal transform (METNN) is defined as

$$\|\mathcal{X}\|_{\mathrm{METNN}}\triangleq\sum_{k=1}^{d}\frac{1}{d}\|\mathcal{X}\|_{\mathrm{ETNN}_k}. \tag{21}$$

Furthermore, for any semi-orthogonal transform $\boldsymbol{T}_k \in \mathbb{R}^{n_k \times \ell_k}$, the enhanced multi-mode element-based tensor nuclear norm with semi-orthogonal transform (METNN$^+$) is defined as

$$\|\mathcal{X}\|_{\text{METNN}^+} \triangleq \sum_{k=1}^{d} \frac{1}{d} \|\mathcal{X}\|_{\text{ETNN}_k^+}. \tag{22}$$

We propose two tensor unclear norms under orthogonal and semi-orthogonal conditions, but the relationship between $\|\mathcal{X}\|_{\text{METNN}}$ and $\|\mathcal{X}\|_{\text{METNN}^+}$ is still not clear. To answer this question, we give the following theorem.

Theorem 2: For $\mathcal{X} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ with any orthogonal transform $\boldsymbol{T}_k \in \mathbb{R}^{n_k \times n_k}$ and semi-orthogonal transform $\boldsymbol{D}_k \in \mathbb{R}^{n_k \times \ell_k}$ that satisfies $\boldsymbol{D}_k = \boldsymbol{T}_k(:, 1:\ell_k)$. The METNN with orthogonal transform $\boldsymbol{T}_k$ and METNN$^+$ with semi-orthogonal transform $\boldsymbol{D}_k$ show the relationship such that

$$\|\mathcal{X}\|_{\text{METNN}^+} \leq \|\mathcal{X}\|_{\text{METNN}}. \tag{23}$$

The proof of Theorem 2 can be found in the supplementary material.

In Definitions 14 and 15, the connection between single-mode and multi-mode is established by the permuting operation applied to $\mathcal{X}$. The following theorem emphasizes that this connection can be established under the transformed domain, which is equivalent to the connection under the original domain, i.e., the permuting operation applied to $\Phi(\mathcal{X}) := \mathcal{C}$.

Theorem 3: For any dth-order tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$, we assume that it can be written as $\mathcal{X} = \sum_{k=1}^{d} \frac{1}{d} \mathcal{X}_k$. Let $\mathcal{C} = \Phi(\mathcal{X})$, if $\mathcal{X}_k = \mathcal{U}_k \Diamond_k \mathcal{S}_k \Diamond_k \mathcal{V}_k^{\mathtt{H}_k}$, then we have $\mathcal{C} = \sum_{k=1}^{d} \frac{1}{d} \mathcal{C}_k$ and $\mathcal{C}_k = \Phi(\mathcal{X}_k)$. It can be further verified that

$$\begin{aligned} \|\mathcal{X}\|_{\text{METNN}} &= \sum_{k=1}^{d} \frac{1}{d} \|\mathcal{X}\|_{\text{ETNN}_k} \\ &= \sum_{k=1}^{d} \frac{1}{d} \|\text{Hbdiag}\left(\mathcal{C}_{\langle k \rangle}\right)\|_*. \end{aligned} \tag{24}$$

The proof of Theorem 3 can be found in the supplementary material.

V. Tensor Exact Recovery Guarantee Based on Element-Based Tensor SVD

In this section, we first present the LRTC model based on the METNN and METNN$^+$, in the meanwhile proving the exact recovery guarantee of the model. Then, we develop an efficient algorithm with the ADMM framework and analyze the convergence behavior and the computational complexity.

A. Low-Rank Tensor Completion Model

Let $\mathcal{O} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ be the observation tensor, then we assume that the ground truth $\mathcal{X}^* \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ takes a low-rank structure with tensor-element multi-rank R. The observed index set of $\mathcal{O}$ is denoted by $\Omega = \{(i_1, i_2, \cdots, i_d) \mid \delta_{i_1 i_2 \cdots i_d} = 1\}$ in which the independent and identically distributed (i.i.d.) binary variable $\delta_{i_1 i_2 \cdots i_d}$ follows a Bernoulli distribution with a probability of ρ. The LRTC model allows one to reconstruct $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ from a very limited number of observations, mainly due to its low-rank structure. Formally, the LRTC model can be formulated as follows,

$$\min_{\mathcal{X}} \text{rank}(\mathcal{X}), \text{ s .t. } \mathcal{X}_\Omega = \mathcal{O}_\Omega. \tag{25}$$

However, minimizing the rank function is NP-hard. We suggest the proposed METNN replace the rank function to find feasible low-rank structures. Therefore, the model (25) can be rather expressed as

$$\min_{\mathcal{X}} \|\mathcal{X}\|_{\text{METNN}}, \text{ s.t. } \mathcal{X}_\Omega = \mathcal{O}_\Omega. \tag{26}$$

The model (26) is built in the invertible transform framework, and then we will prove its exact recovery guarantee in Theorem 4. In the case that the transform is non-invertible, as theoretical properties may be compromised, it is possible to improve performance and reduce computational complexity if $\boldsymbol{T}_k$ is a semi-orthogonal transform. As a result, the model (25) can be expressed as

$$\min_{\mathcal{X}} \|\mathcal{X}\|_{\text{METNN}^+}, \text{ s.t. } \mathcal{X}_\Omega = \mathcal{O}_\Omega. \tag{27}$$

B. Exact Recovery Guarantee

We establish the exact recovery guarantee in this part. We first introduce some basic definition used for the exact recovery guarantee.

Definition 16 (Tensor incoherence conditions): For $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ suppose that the mode-k elt-SVD is $\mathcal{X}_k = \mathcal{U}_k \Diamond_k \mathcal{S}_k \Diamond_k \mathcal{V}_k^{\mathtt{H}_k}$ with tensor-element multi-rank $\text{rank}_{\text{tm}}(\mathcal{X}) = R$. Let $\mathcal{X} = \sum_{k=1}^{d} \frac{1}{d} \mathcal{X}_k$ and $M_k = \prod_{i \neq k, k+1}^{d} n_i$. Then, for any k, the tensor incoherence conditions with parameter $\mu > 1$ are given by

$$\max_{i_k = 1, \cdots, n_l,\ l \neq k+1} \|\mathcal{U}_{\langle k \rangle} \Diamond_1 \mathring{e}_k^{(i_k)}\|_F^2 \leq \frac{\mu \sum_{i=1}^{M_k} R(k,i)}{n_k M_k}, \tag{28}$$

$$\max_{i_k = 1, \cdots, n_l,\ l \neq k} \|\mathcal{V}_{\langle k \rangle} \Diamond_1 \mathring{e}_{k+1}^{(i_{k+1})}\|_F^2 \leq \frac{\mu \sum_{i=1}^{M_k} R(k,i)}{n_{k+1} M_k}, \tag{29}$$

where $\mathcal{U}_{\langle k \rangle}$ and $\mathcal{V}_{\langle k \rangle}$ are mode-k permutation of $\mathcal{U}_k$ and $\mathcal{V}_k$, respectively; $\mathring{e}_k^{(i_k)}$ is the standard dth-order tensor basis whose size is $n_1 \times \cdots \times n_k \times 1 \times n_{k+2} \times \cdots \times n_d$ with its $(i_1, \cdots, i_k, 1, i_{k+2}, \cdots, i_d)$-th entry be 1 and be 0 otherwise, and the $\mathring{e}_{k+1}^{(i_{k+1})} = (\mathring{e}_k^{(i_k)})^{\mathtt{H}_k}$.

Theorem 4: For $\mathcal{X}^* \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ with fixed orthonormal matrix $\boldsymbol{T}_k \in \mathbb{R}^{n_k \times n_k}$ and the mode-k tensor-element SVD of $\mathcal{X}_k = \mathcal{U}_k \Diamond_k \mathcal{S}_k \Diamond_k \mathcal{V}_k^{\mathtt{H}_k}$ with tensor-element multi-rank $\text{rank}_{\text{tm}}(\mathcal{X}) = R$, they are satisfied with $\mathcal{X}^* = \sum_{k=1}^{d} \frac{1}{d} \mathcal{X}_k$, $M_k = \prod_{i \neq k, k+1}^{d} n_i$, $n_k^{(1)} = \max(n_k, n_{k+1})$, $n_k^{(2)} = \min(n_k, n_{k+1})$, and $\dot{n} = \max(n_1^{(1)} M_1, \cdots, n_d^{(1)} M_d)$. Suppose that the indices set $\Omega \sim Ber(\rho)$ with $|\Omega| = m$ and the tensor incoherence conditions (28)-(29) hold. Then, there exist universal constants $c_1, c_2, c_3 > 0$ such that $\mathcal{X}$ is the unique solution to (26) with probability at least $1 - c_1 \dot{n}^{c_2}$, provided that

$$m \geq \frac{c_3 \mu}{d} \sum_{k=1}^{d} \sum_{i=1}^{M_k} R(k,i)\, n_k^{(1)} \log\left(n_k^{(1)} M_k\right). \tag{30}$$

The proof of Theorem 4 can be found in the supplementary material.

C. Proposed Algorithm

Based on the Definition 14, we introduce the auxiliary variables $\mathcal{Z}_k$ to solve the proposed models, for $k \in [d]$, and the models (26) and (27) can be reformulated as

$$\begin{aligned}&\min_{\mathcal{X},\mathcal{C},\mathcal{Z}_k,T_k} \sum_{k=1}^{d} \frac{1}{d}\|\text{Hbdiag}\left(\mathcal{Z}_k,k\right)\|_*, \quad \text{s.t. } \mathcal{X}_\Omega = \mathcal{O}_\Omega,\\ &\mathcal{C}_k = \mathcal{Z}_k, \mathcal{X} = \mathcal{C} \times_1 \boldsymbol{T}_1 \times_2 \cdots \times_d \boldsymbol{T}_d, \boldsymbol{T}_k^\top \boldsymbol{T}_k = \boldsymbol{I}, \end{aligned} \tag{31}$$

where $\mathcal{C}_k = \Phi\left(\mathcal{X}_k\right)$ (see Theorem 3). The model (31) serves as a general format representing both the model (26) and model (27), where it is equivalent to the model (26) when the $\boldsymbol{T}_k$ is orthogonal transform, and the model (27) when the $\boldsymbol{T}_k$ is semi-orthogonal transform, respectively. To convert problem (31) into an unconstrained problem, we introduce the indicator functions as

$$\Phi\left(\mathcal{X}\right) = \begin{cases} 0, & \mathcal{X}_\Omega = \mathcal{O}_\Omega, \\ +\infty, & otherwise. \end{cases} \tag{32}$$

$$\Psi\left(\boldsymbol{T}_k\right) = \begin{cases} 0, & \boldsymbol{T}_k^\top \boldsymbol{T}_k = \boldsymbol{I}, \\ +\infty, & otherwise. \end{cases} \tag{33}$$

Based on the ADMM algorithm [63], the augmented Lagrangian function of (31) is

$$\begin{aligned}&\mathcal{L}_\beta\left(\mathcal{X},\mathcal{C},\mathcal{Z}_k,\boldsymbol{T}_k,\Gamma_k,\Lambda\right) \\ &= \sum_{k=1}^{d} \frac{1}{d}\left\{\|\text{Hbdiag}\left(\mathcal{Z}_k,k\right)\|_* + \frac{\beta}{2}\|\mathcal{C}_k - \mathcal{Z}_k + \frac{\Gamma_k}{\beta}\|_F^2\right\} \\ &+ \frac{\beta}{2}\|\mathcal{X} - \mathcal{C}\times_1 \boldsymbol{T}_1 \times_2 \cdots \times_d \boldsymbol{T}_d + \frac{\Lambda}{\beta}\|_F^2 + \Phi\left(\mathcal{X}\right) + \Psi\left(\boldsymbol{T}_k\right),\end{aligned} \tag{34}$$

where $\Gamma_{\langle k\rangle}$ and Λ are the Lagrange multipliers; $k \in [d]$; β is the penalty parameter. Within the framework of ADMM, $\mathcal{X}, \mathcal{C}, \mathcal{Z}_k, \boldsymbol{T}_k, \Gamma_k$, and Λ are alternately updated by

$$\begin{cases} \mathcal{X}^{t+1} = \arg\min_{\mathcal{X}} \mathcal{L}_\beta\left(\mathcal{X},\mathcal{C}^t,\mathcal{Z}_k^t,\boldsymbol{T}_k^t,\Gamma_k^t,\Lambda^t\right), \\ \mathcal{C}^{t+1} = \arg\min_{\mathcal{C}} \mathcal{L}_\beta\left(\mathcal{X}^{t+1},\mathcal{C},\mathcal{Z}_k^t,\boldsymbol{T}_k^t,\Gamma_k^t,\Lambda^t\right), \\ \mathcal{Z}_k^{t+1} = \arg\min_{\mathcal{Z}_k} \mathcal{L}_\beta\left(\mathcal{X}^{t+1},\mathcal{C}^{t+1},\mathcal{Z}_k,\boldsymbol{T}_k^t,\Gamma_k^t,\Lambda^t\right), \\ \boldsymbol{T}_k^{t+1} = \arg\min_{\boldsymbol{T}_k} \mathcal{L}_\beta\left(\mathcal{X}^{t+1},\mathcal{C}^{t+1},\mathcal{Z}_k^{t+1},\boldsymbol{T}_k,\Gamma_k^t,\Lambda^t\right), \\ \Gamma_k^{t+1} = \Gamma_k^t + \beta\left(\mathcal{C}_k^{t+1} - \mathcal{Z}_k^{t+1}\right), \\ \Lambda^{t+1} = \Lambda^t + \beta\left(\mathcal{X}^{t+1} - \mathcal{C}^{t+1} \times_1 \cdots \times_d \boldsymbol{T}_k^{t+1}\right). \end{cases} \tag{35}$$

1) The $\mathcal{X}$ subproblem is

$$\mathcal{X}^{t+1} = \arg\min_{\mathcal{X}} \frac{\beta}{2}\|\mathcal{X} - \mathcal{C}^t \times_1 \cdots \times_d \boldsymbol{T}_d^t + \frac{\Lambda^t}{\beta}\|_F^2 + \Phi\left(\mathcal{X}\right). \tag{36}$$

The closed-form solution of (36) is

$$\mathcal{X}^{t+1} = \left(\mathcal{C}^{t+1} \times_1 \boldsymbol{T}_1^t \times_2 \cdots \times_d \boldsymbol{T}_d^t - \frac{\Lambda^t}{\beta}\right)_{\Omega^C} + \mathcal{O}_\Omega, \tag{37}$$

where Ω^C is the complementary set of Ω.

2) The $\mathcal{C}$ subproblem is

$$\begin{aligned}\mathcal{C}^{t+1} = \arg\min_{\mathcal{C}} &\sum_{k=1}^{d} \frac{1}{d}\left\{\frac{\beta}{2}\|\mathcal{C}_k - \mathcal{Z}_k^t + \frac{\Gamma_k^t}{\beta}\|_F^2\right\} \\ &+ \frac{\beta}{2}\|\mathcal{X}^{t+1} - \mathcal{C} \times_1 \boldsymbol{T}_1^t \times_2 \cdots \times_d \boldsymbol{T}_k^t + \frac{\Lambda^t}{\mu}\|_F^2.\end{aligned} \tag{38}$$

The closed-form solution of (38) is

$$\mathcal{C}^{t+1} = \frac{\sum_{k=1}^{d} \frac{1}{d}\left(\mathcal{Z}_k^t - \frac{\Gamma_k^t}{\beta}\right) + \mathcal{H}^t}{2}, \tag{39}$$

where $\mathcal{H}^t = \left(\mathcal{X}^{t+1} + \frac{\Lambda^t}{\beta}\right) \times \boldsymbol{T}_1^{t\top} \times_2 \cdots \times_d \boldsymbol{T}_d^{t\top}$.

3) The $\mathcal{Z}_k, k \in [d]$ subproblems are

$$\begin{aligned}\mathcal{Z}_k^{t+1} &= \arg\min_{\mathcal{Z}_k} \|\text{Hbdiag}\left(\mathcal{Z}_k,k\right)\|_* + \frac{\beta}{2}\|\mathcal{C}_k^{t+1} - \mathcal{Z}_k + \frac{\Gamma_k}{\beta}\|_F^2 \\ &= \arg\min_{\mathcal{Z}_k} \frac{\beta}{2}\|\mathcal{C}_k^{t+1} - \mathcal{Z}_k + \frac{\Gamma_k}{\beta}\|_F^2 \\ &+ \sum_{i_1=1}^{\ell_1} \cdots \sum_{i_{k-1}=1}^{\ell_{k-1}} \sum_{i_{k+2}=1}^{\ell_{k+2}} \cdots \sum_{i_d=1}^{\ell_d} \|\mathcal{Z}^{(i_1,\cdots,i_{k-1},:,:,i_{k+2},\cdots,i_d)}\|_*.\end{aligned} \tag{40}$$

Then, the problem (40) is equivalent to solving $\prod_{i\neq k,k+1}^{d} \ell_i$ individual subproblems, i.e.,

$$\begin{aligned}&\arg\min_{\boldsymbol{Z}} \|\boldsymbol{Z}^{(i_1,\cdots,i_{k-1},i_{k+2},\cdots,i_d)}\|_* \\ &+ \frac{\beta}{2}\|\boldsymbol{Z}^{(i_1,\cdots,i_{k-1},i_{k+2},\cdots,i_d)} - \boldsymbol{D}^{(i_1,\cdots,i_{k-1},i_{k+2},\cdots,i_d)}\|_F^2,\end{aligned} \tag{41}$$

where $\boldsymbol{Z}^{(i_1,\cdots,i_{k-1},i_{k+2},\cdots,i_d)} = \mathcal{Z}(i_1,\cdots,i_{k-1},:,:,i_{k+2},\cdots,i_d)$, $\boldsymbol{D}^{(i_1,\cdots,i_{k-1},i_{k+2},\cdots,i_d)} = \mathcal{D}(i_1,\cdots,i_{k-1},:,:,i_{k+2},\cdots,i_d) \in \mathbb{R}^{n_k \times n_{k+1}}$, and $\mathcal{D}_k = \mathcal{C}_k^{t+1} + \frac{\Gamma_k^t}{\beta}$. The closed-form solution to the variable $\boldsymbol{Z}$ is

$$\boldsymbol{Z}^{(i_1,\cdots,i_{k-1},i_{k+2},\cdots,i_d)} = \text{SVT}_{\frac{1}{\beta}}(\boldsymbol{D}^{(i_1,\cdots,i_{k-1},i_{k+2},\cdots,i_d)}), \tag{42}$$

where $\text{SVT}_\tau(\cdot)$ is the singular value thresholding (SVT) operator [64] that associated with the threshold value τ.

4) The $\boldsymbol{T}_k$ subproblems are as follows:

$$\begin{aligned}\boldsymbol{T}_k^{t+1} &= \arg\min_{T_k} \frac{\beta}{2}\|\mathcal{X}^{t+1} - \left[\mathcal{C}^{t+1};\boldsymbol{T}_{1:k-1}^t,\boldsymbol{T}_k,\boldsymbol{T}_{k+1:d}^t\right] \\ &+ \frac{\Lambda^t}{\beta}\|_F^2 + \Psi\left(\boldsymbol{T}_k\right) \\ &= \arg\min_{\boldsymbol{T}_k} \frac{\beta}{2}\|\left[\mathcal{C}^{t+1};\boldsymbol{T}_{1:k-1}^t,\boldsymbol{T}_{k+1:d}^t\right] - \left(\mathcal{X}^{t+1} + \frac{\Lambda^t}{\beta}\right) \\ &\times_k \boldsymbol{T}_k^\top\|_F^2 + \Psi\left(\boldsymbol{T}_k\right),\end{aligned} \tag{43}$$

where $\left[\mathcal{C}^{t+1};\boldsymbol{T}_{1:k-1}^t,\boldsymbol{T}_k,\boldsymbol{T}_{k+1:d}^t\right]$ is another symbol of the mode-k product, i.e. $[\mathcal{C}^{t+1};\ \boldsymbol{T}_{1:k-1}^t,\boldsymbol{T}_k,\boldsymbol{T}_{k+1:d}^t] = \mathcal{C}^{t+1} \times \boldsymbol{T}_1^t \times_2 \cdots \times_d \boldsymbol{T}_d^t$. We let $\mathcal{P} = \mathcal{X}^{t+1} + \frac{\Lambda^t}{\beta}$ and

Algorithm 1 ADMM algorithm for the proposed models.

Input: The observed tensor $\mathcal{O} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$, the index Ω, the parameter of transform matrix $\boldsymbol{\ell} = (\ell_1, \cdots, \ell_d) \in \mathbb{R}^d$, the parameter β and γ, and the maximum iteration $t_{max} = 1000$.

Output: The reconstructed tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$.

Initialization: The iteration $t = 0$, $\mathcal{X}^0$, $\mathcal{C}^0$, $\mathcal{Z}^0$, and $\boldsymbol{T}_k^0$.
while not converged and $t < t_{max}$ **do**
  Update $\mathcal{X}^{t+1}$ via Eq. (37);
  Update $\mathcal{C}^{t+1}$ via Eq. (39);
  Update $\mathcal{Z}^{t+1}$ via Eq. (42);
  Update $\boldsymbol{T}_k^{t+1}$ via Eq. (45);
  Update Γ_k^{t+1} and Λ^{t+1} via Eq. (35);
  Let $\beta^{t+1} = \gamma\beta^t$ and $t = t + 1$;
  Check the convergence conditions

$$\left\|\mathcal{X}^{t+1} - \mathcal{X}^t\right\|_F / \left\|\mathcal{X}^t\right\|_F \leq 10^{-4};$$

end while

$\mathcal{Q} = \left[\mathcal{C}; \boldsymbol{T}_{1:k-1}^t, \boldsymbol{T}_{k+1:d}^t\right]$, and then the equation (43) can be converted into the following formulation:

$$\begin{aligned}\boldsymbol{T}_k^{t+1} &= \arg\min_{\boldsymbol{T}_k} \frac{\beta}{2}\|\boldsymbol{Q}_{\langle k\rangle} - \boldsymbol{T}_k^\top \boldsymbol{P}_{\langle k\rangle}\|_F^2 + \Psi\left(\boldsymbol{T}_k\right) \\ &= \arg\max_{\boldsymbol{T}_k} \operatorname{Tr}\left(\left(\beta \boldsymbol{Q}_{\langle k\rangle} \boldsymbol{P}_{\langle k\rangle}^\top\right) \boldsymbol{T}_k\right) - \Psi\left(\boldsymbol{T}_k\right). \end{aligned} \quad (44)$$

This is an orthogonal Procrustes problem [65], and the closed-form solution of (44) is

$$\boldsymbol{T}_k^{\top t+1} = \boldsymbol{U}_k^{t+1} \boldsymbol{V}_k^{\top t+1}, \quad (45)$$

where $\boldsymbol{U}_k^{t+1}$, $\boldsymbol{S}_k^{t+1}$, and $\boldsymbol{V}_k^{t+1}$ are results of SVD on $\beta \boldsymbol{Q}_{\langle k\rangle} \boldsymbol{P}_{\langle k\rangle}^\top$. The orthogonality or semi-orthogonality of $\boldsymbol{T}_k^{t+1}$ depends on the size of the initially given $\boldsymbol{T}_k^0$. For initially given $\boldsymbol{T}_k^0 \in \mathbb{R}^{n_k \times \ell_k}$, the $\boldsymbol{T}_k^0$ is orthogonal and is the solution of model (26) when $\ell_k = n_k$. The $\boldsymbol{T}_k^{t+1}$ is semi-orthogonal and is the solution of model (27) when $0 < \ell_k < n_k$.

D. Computational Complexity Analysis

In this section, we analyze the computational complexity of the proposed models in Algorithm 1. For $\mathcal{X} \in \mathbb{R}^{n \times n \times \cdots \times n}$ and transform matrix $\boldsymbol{T}_k \in \mathbb{R}^{n \times \bar{\ell}}$, the main computational complexity at each iteration for the proposed algorithm can be concluded by updating $\mathcal{X}$, updating $\mathcal{C}$, updating $\mathcal{Z}_k$, and updating $\boldsymbol{T}_k$, which cost $\mathcal{O}\left(\sum_{k=1}^d \bar{\ell}^k n^{d-k+1}\right)$, $\mathcal{O}\left(\sum_{k=1}^d \bar{\ell}^k n^{d-k+1}\right)$, $\mathcal{O}(d\bar{\ell}^{d+1})$, and $\mathcal{O}\left(N\bar{\ell} n^d + dn^2\bar{\ell} + dn\bar{\ell}^2\right)$, respectively. Therefore, the overall computational complexity at each iteration of Algorithm 1 is $\mathcal{O}\left(\sum_{k=1}^d \bar{\ell}^k n^{d-k+1} + d\bar{\ell}^{d+1} + d\bar{\ell} n^d\right)$. In addition, the computational complexities of the algorithms of HaLRTC, TNN, UTNN, HTNN, and WSTNN methods are $\mathcal{O}(dn^{d+1})$, $\mathcal{O}(n^d \log(n^{d-2}) + n^{d+1})$, $\mathcal{O}(n^{2d-2} + n^{d+1})$, $\mathcal{O}(n^{2d-2} + n^{d+1})$, and $\mathcal{O}((\frac{d(d+1)}{2})\ (n^d \log(n^{d-2}) + n^{d+1}))$, respectively.

E. Convergence Analysis

In this section, we provide the theoretical convergence of our algorithm. The architecture of proof is to refer to [66], where the algorithm satisfies the following conditions:

1) The sequence $(\mathcal{X}^t, \mathcal{C}^t, \mathcal{Z}_k^t, \boldsymbol{T}_k^t, \Gamma_k^t, \Lambda^t)_{t\in\mathbb{N}}$ satisfies sufficient decrease condition;
2) The sequence $(\mathcal{X}^t, \mathcal{C}^t, \mathcal{Z}_k^t, \boldsymbol{T}_k^t, \Gamma_k^t, \Lambda^t)_{t\in\mathbb{N}}$ is bounded;
3) The $\mathcal{L}_\mu(\mathcal{X}, \mathcal{C}, \mathcal{Z}_k, \boldsymbol{T}_k, \Gamma_k, \Lambda)$ is a proper lower semi-continuous function and has the Kurdyka-Łojasiewicz (K-Ł) property at $(\mathcal{X}^t, \mathcal{C}^t, \mathcal{Z}_k^t, \boldsymbol{T}_k^t, \Gamma_k^t, \Lambda^t)_{t\in\mathbb{N}}$;
4) The $\mathcal{L}_\mu(\mathcal{X}, \mathcal{C}, \mathcal{Z}_k, \boldsymbol{T}_k, \Gamma_k, \Lambda)$ satisfies subgradient bound condition.

Theorem 5: Suppose the sequence $(\mathcal{X}^t, \mathcal{C}^t, \mathcal{Z}_k^t, \boldsymbol{T}_k^t, \Gamma_k^t, \Lambda^t)_{t\in\mathbb{N}}$ is generated by the proposed algorithm, and it satisfies the conditions 1-4. Then, the sequence converges to a critical point of $\mathcal{L}_\mu(\mathcal{X}, \mathcal{C}, \mathcal{Z}_k, \boldsymbol{T}_k, \Gamma_k, \Lambda)$.

The proof of Theorem 5 can be found in the supplementary material.

VI. Numerical Experiments

In this section, we conduct experiments to verify the effectiveness of the proposed methods for LRTC on both synthetic tensors and real-world data. For numerical experiments on synthetic tensors, we investigate the probability of successful recovery with simultaneous varying of the tensor-element rank r and the sampling rate ρ. For numerical experiments on real-world data, we investigate the performance of the proposed METNN and METNN$^+$ on fourth-order tensor data (i.e., color videos and traffic data), fifth-order tenor data (i.e., light field images), and sixth-order tensor data (i.e., light field videos). Especially, we compare our models with several state-of-the-art methods, including a Tucker rank baseline method–HaLRTC [67], a third-order t-SVD baseline method–TNN [44], a third-order unitary transform-based method–UTNN [50], dth-order TNN method–HTNN [60] and WSTNN [57]. For third-order tensor methods, we use dimensional merging to obtain a third-order tensor $\tilde{\mathcal{X}} \in \mathbb{R}^{n_1 \times n_2 \times (n_3 \cdots n_d)}$[2] as input. All numerical experiments are implemented in Windows 10 64-bit and MATLAB R2022a on a desktop computer with an Intel(R) Core(TM) i9-12900 CPU at 2.40 GHz with 64GB memory of RAM.

A. Simulations

To demonstrate Theorem 4, we conduct a series of independent numerical experiments aimed at recovering randomly missing tensors. For each independent experiment, we generate a random tensor of size $30 \times 30 \times 30 \times 30$ with Gaussian distribution. Subsequently, we perform the mode-k tensor-element SVD of it to get $\mathcal{X}$ with tensor-element rank $(r, r, r, r) \in \mathbb{R}^4$. We randomly sample $\rho n_1 n_2 n_3 n_4$ entries from $\mathcal{X}$ and then try to recover the missing tensors using various methods: TNN, HTNN, WSTNN, and METNN. The success rate is defined as the ratio of the number of successful recoveries to the total number of attempts. A test is deemed to be successful if the relative

[2] $\tilde{\mathcal{X}}$ can be obtained by the MATLAB command reshape($\mathcal{X}$, $[n_1, n_2, \text{prod}(n_3 \cdots n_d)]$)

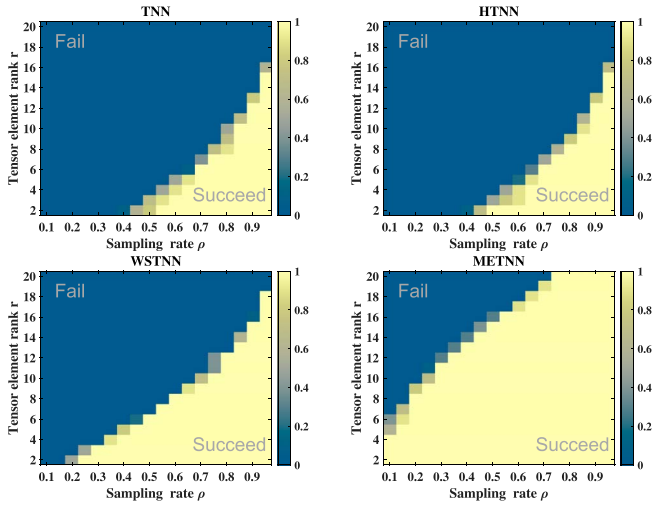


Fig. 2. The success rates for synthetic data with a varying tensor-element rank r and varying SRs ρ.

squared error between the recovered tensor $\mathcal{X}^*$ and the ground-truth tensor $\mathcal{X}$ satisfies $\|\mathcal{X}^* - \mathcal{X}\|_F^2 / \|\mathcal{X}\|_F^2 \leq 10^{-3}$. We repeat our experiments 10 times and calculate the success rate. Fig. 2 reports the success rate of TNN, HTNN, WSTNN, and METNN methods at different sampling rates (SRs) ρ and tensor-element rank r. A larger yellow area indicates a higher recovery capability for the respective method. The observations from Fig. 2 reveal that the METNN method exhibits a reduced sampling requirement for successfully recovering the underlying tensor, in comparison to the TNN, HTNN, and WSTNN methods. These simulation experiments further support the findings presented in Theorem 4.

B. Experiments on Color Videos

We evaluate the performance of METNN and METNN$^+$ on seven videos from the NTT database[3], including *Rhino*, *Bird*, *Horse*, *Fox*, *Board*, *Ski*, and *Flight*. In our experiments, we select the first 30 frames of each video, and each video is resized to $144 \times 176 \times 3 \times 30$. For the each data, SRs are set to be 0.05, 0.10, 0.15, and 0.20, respectively.

Table II displays the numerical evaluation metrics of various methods at different SRs. From Table II, it can be seen that the proposed methods METNN and METNN$^+$ achieve significant advantages compared to other methods. With the increase in SRs, the advantages being established by proposed methods are expanding, whether in terms of PSNR, SSIM, or MSE. Fig. 3 presents the recovered results, zoom-in regions, and the corresponding residual images obtained by the HaLRTC, TNN, UTNN, HTNN, WSTNN, METNN, and METNN$^+$ methods at SR=0.2. Observing Fig. 3, it becomes evident that our proposed methods, METNN and METNN$^+$, achieve superior results in both edge profile and fine details compared to the other approaches. More visual results on videos *Bird*, *Horse*, *Fox*, *Board*, and *Ski* can be found in the supplementary material.

[3]The data is available at http://www.brl.ntt.co.jp/people/akisato/saliency3.html.

C. Experiments on Light Field Images

To evaluate the performance of METNN and METNN$^+$, we select some light field images with 9×9 views from the Light Field Database[4], including *Dish*, *Greek*, *Kitchen*, *Medieval*, *Table*, and *Town*. For each light field image, it is resized to $128 \times 128 \times 3 \times 9 \times 9$ in our experiments. For each data, SRs are set to be 0.05, 0.10, 0.15, and 0.20, respectively.

Table III reports the PSNR, SSIM, and MSE of all methods on six light field images with different SRs. The proposed methods, benefiting from the exclusion of the unfolding operation and the incorporation of the direction prior, outperform HaLRTC, TNN, UTNN, HTNN, and WSTNN in terms of PSNR, SSIM, and MSE. Fig. 4 shows the visual restoration results by different methods, where SR is 0.10 at *Dish* and *Greek*. From Fig. 4, the visual results obtained by the proposed methods are outperformed the comparison model on different light field data. More visual results about *Kitchen*, *Medieval*, *Table*, and *Town* can be found in the supplementary material.

D. Experiments on Light Field Videos

To evaluate the performance of METNN and METNN$^+$, we select two light field videos with 3×3 views from [68][5]. In our experiments, we select the first 20 frames of each video, and each video is resized to $256 \times 256 \times 3 \times 20 \times 3 \times 3$ in our experiments. For each data, SRs are set to be 0.05, 0.10, 0.15, and 0.20, respectively.

Table IV provides a comprehensive overview of quantitative evaluation metrics about different SRs. Remarkably, the proposed methods (METNN and METNN$^+$) consistently demonstrate substantial advantages over their counterparts. In Fig. 5, we present a detailed visual assessment encompassing restoration results, magnified regions, and corresponding residual images obtained through HaLRTC, TNN, UTNN, HTNN, WSTNN, METNN, and METNN$^+$ under SR = 0.2. Similar to color videos and light field images, Fig. 5 reflects the consistently excellent performance of the proposed methods METNN and METNN$^+$ in light field videos, especially in preserving edge contours and complex fine details.

E. Experiments on Traffic Data

To verify the effectiveness of the proposed methods on traffic data, we choose two traffic data sets collected by the California department of transportation through their Performance Measurement System (PeMS)[6] with a 5-min resolution. One dataset covers 8 weeks with 200 sensors (referred to as PeMS-8W), whose size is $200 \times 12 \times 24 \times 56$, and the other covers 12 weeks with 100 sensors (referred to as PeMS-12W), whose size is $100 \times 12 \times 24 \times 82$. To evaluate the restoration results for traffic data, we used three metrics [69] including MSE, MAPE, RMSE. For each data, SRs are set to be 0.05, 0.10, 0.15, and 0.20, respectively. The lower MSE, MAPE, and RMSE values

[4]The data is available at https://lightfield-analysis.uni-konstanz.de/.

[5]The data is available at https://resources.mpi-inf.mpg.de/LightFieldVideo/index.html.

[6]The data is available at https://zenodo.org/records/3939793.

TABLE II
EVALUATION INDICES OF RESTORATION RESULTS BY DIFFERENT METHODS ON COLOR VIDEOS UNDER DIFFERENT SRS

		SR=0.05			SR=0.10			SR=0.15			SR=0.20		
Data	Method	PSNR	SSIM	MSE	PSNR	SSIM	MSE	PSNR	SSIM	MSE	PSNR	SSIM	MSE
Rhino	HaLRTC	17.467	0.370	0.297	21.827	0.525	0.177	23.288	0.591	0.148	24.618	0.653	0.127
	TNN	23.809	0.559	0.139	25.888	0.662	0.109	27.164	0.723	0.094	28.292	0.770	0.083
	UTNN	24.131	0.562	0.134	26.213	0.666	0.105	27.651	0.735	0.088	29.130	0.796	0.074
	HTNN	23.943	0.567	0.137	26.322	0.687	0.104	27.948	0.761	0.086	29.430	0.817	0.073
	WSTNN	25.366	0.662	0.117	27.576	0.765	0.090	29.303	0.831	0.074	30.888	0.877	0.061
	METNN	26.237	**0.684**	0.104	28.621	0.785	0.079	30.998	0.869	0.060	32.767	**0.910**	0.049
	METNN$^+$	**26.465**	0.678	**0.101**	**29.083**	**0.786**	**0.075**	**31.347**	**0.873**	**0.058**	**33.098**	0.908	**0.047**
Bird	HaLRTC	15.271	0.317	0.402	19.889	0.473	0.236	21.939	0.585	0.186	23.750	0.680	0.151
	TNN	24.138	0.694	0.145	27.081	0.816	0.103	28.897	0.865	0.084	30.468	0.897	0.070
	UTNN	24.766	0.725	0.135	27.592	0.830	0.098	29.594	0.879	0.078	31.388	0.912	0.063
	HTNN	24.170	0.695	0.144	27.243	0.821	0.101	29.228	0.872	0.081	30.960	0.906	0.066
	WSTNN	26.043	0.815	0.117	29.363	0.901	0.080	31.699	0.935	0.061	33.758	0.956	0.049
	METNN	26.200	0.799	0.115	30.215	0.906	0.072	33.070	**0.945**	0.052	35.482	0.965	0.039
	METNN$^+$	**27.006**	**0.816**	**0.105**	**31.293**	**0.914**	**0.064**	**33.883**	**0.945**	**0.047**	**36.227**	**0.968**	**0.036**
Horse	HaLRTC	16.855	0.436	0.312	22.428	0.588	0.165	24.368	0.657	0.132	26.014	0.718	0.109
	TNN	25.387	0.629	0.118	28.169	0.756	0.086	29.877	0.819	0.071	31.314	0.862	0.060
	UTNN	25.994	0.656	0.110	28.674	0.773	0.081	30.439	0.836	0.066	31.977	0.879	0.055
	HTNN	25.435	0.631	0.118	28.357	0.764	0.084	30.217	0.830	0.068	31.829	0.875	0.057
	WSTNN	27.141	**0.771**	0.097	30.102	0.861	0.069	32.145	0.907	0.055	33.901	0.935	0.045
	METNN	27.716	0.745	0.089	30.715	0.851	0.063	33.108	0.909	0.048	35.083	**0.941**	**0.038**
	METNN$^+$	**28.436**	0.763	**0.082**	**31.544**	**0.862**	**0.057**	**33.331**	**0.911**	**0.047**	**35.129**	0.936	**0.038**
Fox	HaLRTC	11.408	0.240	0.357	21.763	0.671	0.110	24.075	0.750	0.085	25.929	0.805	0.069
	TNN	24.296	0.689	0.085	27.267	0.775	0.062	29.172	0.827	0.050	30.747	0.863	0.042
	UTNN	25.732	0.741	0.075	28.960	0.829	0.053	31.430	0.881	0.040	33.660	0.918	0.031
	HTNN	24.473	0.695	0.083	28.025	0.798	0.056	30.693	0.863	0.041	32.879	0.904	0.032
	WSTNN	27.420	0.862	0.062	31.241	0.924	0.041	34.042	0.952	0.030	36.339	0.967	0.023
	METNN	29.184	**0.876**	**0.049**	33.579	**0.940**	0.030	36.879	**0.966**	0.021	**39.265**	**0.979**	**0.016**
	METNN$^+$	**29.845**	0.872	**0.049**	**34.331**	0.939	**0.028**	**37.205**	0.965	**0.020**	39.136	0.976	**0.016**
Board	HaLRTC	16.585	0.551	0.268	23.454	0.761	0.121	25.646	0.810	0.094	27.422	0.849	0.077
	TNN	26.251	0.743	0.090	28.960	0.826	0.067	30.631	0.866	0.055	31.934	0.893	0.048
	UTNN	27.026	0.781	0.083	29.760	0.854	0.061	31.642	0.892	0.050	33.218	0.918	0.042
	HTNN	26.293	0.744	0.089	29.150	0.832	0.065	31.061	0.876	0.052	32.561	0.905	0.044
	WSTNN	28.181	0.874	0.073	31.158	0.921	0.053	33.341	0.945	0.042	35.051	0.960	0.035
	METNN	29.296	0.885	0.064	32.625	0.937	0.044	35.141	0.963	0.033	37.131	**0.973**	**0.026**
	METNN$^+$	**29.812**	**0.887**	**0.061**	**33.391**	**0.939**	**0.041**	**35.547**	**0.962**	**0.032**	**37.227**	**0.973**	**0.026**
Ski	HaLRTC	10.033	0.188	0.389	19.139	0.454	0.137	21.531	0.546	0.104	23.141	0.617	0.086
	TNN	21.519	0.419	0.105	23.898	0.540	0.080	25.369	0.623	0.068	26.657	0.691	0.058
	UTNN	22.179	0.443	0.097	24.470	0.572	0.075	26.348	0.680	0.060	28.267	0.771	0.048
	HTNN	21.760	0.436	0.102	24.799	0.601	0.072	27.135	0.722	0.055	29.217	0.808	0.043
	WSTNN	23.834	0.625	0.081	26.588	0.753	0.059	28.822	0.838	0.045	30.862	0.892	0.036
	METNN	**25.530**	0.684	**0.066**	29.249	0.844	0.043	32.095	0.915	0.031	34.522	0.949	0.023
	METNN$^+$	25.477	**0.685**	**0.066**	**29.731**	**0.855**	**0.041**	**32.935**	**0.926**	**0.028**	**35.315**	**0.954**	**0.021**
Flight	HaLRTC	14.734	0.385	0.292	23.349	0.815	0.104	25.062	0.858	0.085	26.626	0.890	0.071
	TNN	25.249	0.794	0.084	28.220	0.856	0.060	30.017	0.890	0.049	31.373	0.911	0.042
	UTNN	26.206	0.826	0.075	29.113	0.882	0.054	31.080	0.913	0.043	32.821	0.934	0.035
	HTNN	25.319	0.796	0.083	28.373	0.859	0.059	30.334	0.895	0.047	31.894	0.918	0.039
	WSTNN	27.927	0.914	0.062	31.095	0.952	0.043	33.378	0.968	0.034	35.307	0.977	0.027
	METNN	28.883	**0.918**	**0.054**	32.619	**0.957**	0.035	35.265	**0.975**	0.026	37.505	0.981	**0.020**
	METNN$^+$	**29.043**	0.912	**0.054**	**33.308**	0.952	**0.033**	**35.859**	**0.975**	**0.025**	**38.053**	**0.982**	**0.020**

indicate better recovery performance. Table V reports the MSE, MAPE, and RMSE of all methods on traffic data with different SRs. The proposed methods outperform HaLRTC, TNN, UTNN, HTNN, and WSTNN in terms of PSNR, SSIM, and MSE.

F. Discussion

1) Parameter Study: In this part, we discuss regularization parameter β and the parameter of transform matrix $\boldsymbol{\ell} = (\ell_1, \ell_2, \cdots, \ell_d) \in \mathbb{R}^d$ in the proposed models. To evaluate the influence of these parameters, we use the color video *Rhino* at SR = 0.10 as an example. Since METNN is a special case of METNN$^+$, i.e., $\ell_k = n_k, k = 1, 2, 3, 4$, we discuss only the parameters of METNN$^+$. Moreover, since $n_3 = 3$ for color video, ℓ_3 is set to n_3 in our experiment.

We first analyze the influence of β on the proposed model. Fig. 6(a) illustrates the PSNR and SSIM values corresponding to different β, where β chosen from the set $\{10^{-3}, 5 \times 10^{-3}, 10^{-2}, 5 \times 10^{-2}, 10^{-1}, 5 \times 10^{-1}, 1, 5, 10, 50, 100\}$. From Fig. 6(a), we are able to observe that PSNR and SSIM values are stable for β falling within $[10^{-1}, 1]$, and the

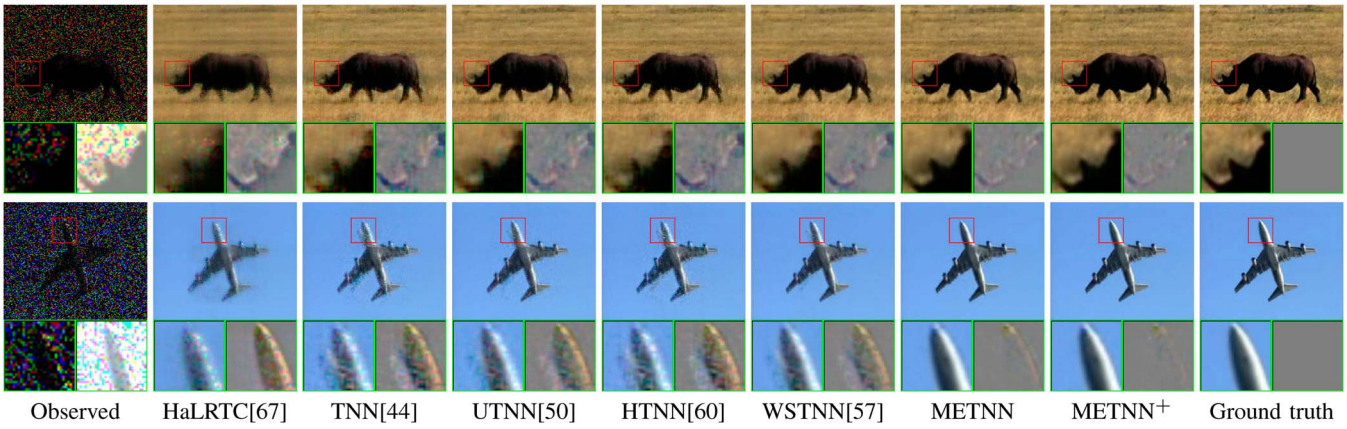


Fig. 3. The selected frame of restoration results by different methods on color videos (i.e., *Rhino* and *Flight*) under SR=0.2.

TABLE III
EVALUATION INDICES OF RESTORATION RESULTS BY DIFFERENT METHODS ON LIGHT FIELD IMAGES UNDER DIFFERENT SRS

		SR=0.05			SR=0.10			SR=0.15			SR=0.20		
Data	Model	PSNR	SSIM	MSE	PSNR	SSIM	MSE	PSNR	SSIM	MSE	PSNR	SSIM	MSE
Dish	HaLRTC	10.226	0.150	0.684	16.673	0.383	0.323	21.840	0.591	0.177	24.078	0.690	0.137
	TNN	28.501	0.825	0.085	31.412	0.898	0.062	33.401	0.930	0.051	35.060	0.949	0.043
	UTNN	29.029	0.844	0.079	32.611	0.923	0.054	35.071	0.954	0.042	37.076	0.970	0.033
	HTNN	29.101	0.843	0.079	32.395	0.917	0.055	34.948	0.950	0.042	36.869	0.966	0.034
	WSTNN	29.321	0.878	0.078	32.761	0.941	0.054	35.223	0.964	0.042	37.342	0.977	0.033
	METNN	30.343	0.892	0.067	34.647	0.959	0.041	**37.893**	**0.980**	**0.029**	40.718	0.990	0.021
	METNN$^+$	**31.220**	**0.906**	**0.061**	**35.679**	**0.964**	**0.037**	37.708	0.976	0.030	**42.232**	**0.993**	**0.018**
Greek	HaLRTC	7.747	0.109	0.733	12.872	0.294	0.407	18.811	0.528	0.206	23.699	0.735	0.117
	TNN	28.858	0.875	0.068	32.743	0.938	0.045	35.019	0.960	0.035	36.919	0.972	0.029
	UTNN	30.906	0.916	0.054	34.946	0.963	0.034	37.644	0.979	0.026	39.979	0.987	0.020
	HTNN	30.131	0.902	0.058	33.897	0.953	0.039	36.964	0.974	0.028	39.141	0.983	0.022
	WSTNN	31.208	0.934	0.053	35.673	0.973	0.033	38.551	0.985	0.024	40.908	0.991	0.019
	METNN	32.936	0.946	0.041	37.547	0.980	0.024	40.709	0.990	0.017	43.541	0.995	0.012
	METNN$^+$	**34.036**	**0.957**	**0.037**	**38.560**	**0.983**	**0.022**	**41.291**	**0.991**	**0.016**	**45.532**	**0.997**	**0.010**
Kitchen	HaLRTC	11.257	0.172	0.686	17.881	0.453	0.320	23.456	0.727	0.169	26.274	0.827	0.122
	TNN	33.895	0.947	0.054	37.135	0.971	0.039	39.574	0.981	0.030	41.419	0.987	0.025
	UTNN	35.643	0.962	0.044	41.204	0.986	0.025	43.861	0.992	0.018	46.553	0.995	0.013
	HTNN	34.560	0.953	0.049	38.648	0.978	0.031	41.842	0.988	0.022	43.883	0.992	0.018
	WSTNN	35.171	0.967	0.047	39.921	0.986	0.029	43.124	0.992	0.021	45.713	0.995	0.016
	METNN	36.142	0.961	0.041	41.285	0.986	0.023	43.957	0.992	0.017	46.384	0.995	0.013
	METNN$^+$	**38.667**	**0.976**	**0.031**	**43.321**	**0.989**	**0.018**	**45.061**	**0.992**	**0.015**	**48.934**	**0.997**	**0.010**
Medieval	HaLRTC	13.777	0.242	0.640	21.112	0.570	0.274	25.604	0.750	0.163	28.176	0.831	0.121
	TNN	34.964	0.948	0.058	38.119	0.971	0.042	40.357	0.981	0.033	42.001	0.986	0.028
	UTNN	36.370	0.961	0.050	41.218	0.985	0.029	43.999	0.991	0.022	45.602	0.994	0.017
	HTNN	35.775	0.955	0.052	39.526	0.978	0.035	42.191	0.987	0.026	43.901	0.991	0.021
	WSTNN	36.209	0.966	0.051	40.582	0.985	0.032	43.311	0.991	0.024	45.482	0.994	0.019
	METNN	37.087	0.962	0.045	41.186	0.984	0.028	43.866	0.991	0.021	45.809	0.994	0.016
	METNN$^+$	**39.017**	**0.975**	**0.036**	**42.916**	**0.988**	**0.023**	**44.537**	**0.991**	**0.019**	**47.727**	**0.996**	**0.013**
Table	HaLRTC	9.021	0.152	0.724	14.353	0.341	0.390	19.914	0.580	0.205	23.542	0.743	0.135
	TNN	30.342	0.907	0.064	33.833	0.950	0.045	36.049	0.967	0.035	37.813	0.976	0.029
	UTNN	32.084	0.937	0.053	37.208	0.975	0.031	40.760	0.987	0.021	43.499	0.992	0.016
	HTNN	30.944	0.917	0.060	34.456	0.957	0.041	37.270	0.975	0.030	39.210	0.983	0.025
	WSTNN	32.482	0.951	0.052	36.842	0.978	0.033	39.828	0.988	0.025	42.284	0.992	0.019
	METNN	33.637	0.949	0.043	38.523	0.980	0.025	41.810	0.990	0.017	44.452	0.994	0.013
	METNN$^+$	**35.652**	**0.961**	**0.036**	**40.372**	**0.983**	**0.021**	**42.080**	**0.988**	**0.017**	**46.855**	**0.996**	**0.010**
Town	HaLRTC	7.964	0.106	0.738	12.782	0.292	0.427	18.171	0.525	0.232	22.903	0.715	0.135
	TNN	29.862	0.896	0.062	33.544	0.947	0.043	35.760	0.965	0.034	37.572	0.975	0.028
	UTNN	32.332	0.936	0.048	37.076	0.975	0.029	40.578	0.987	0.020	43.376	0.993	0.015
	HTNN	30.753	0.912	0.057	33.959	0.952	0.040	36.614	0.971	0.030	38.498	0.980	0.025
	WSTNN	32.503	0.950	0.048	36.548	0.977	0.032	39.532	0.987	0.024	41.993	0.992	0.019
	METNN	33.442	0.946	0.041	37.734	0.978	0.025	40.859	0.989	0.018	43.358	0.993	0.013
	METNN$^+$	**35.306**	**0.959**	**0.033**	**40.297**	**0.985**	**0.019**	**41.864**	**0.989**	**0.016**	**46.150**	**0.996**	**0.010**

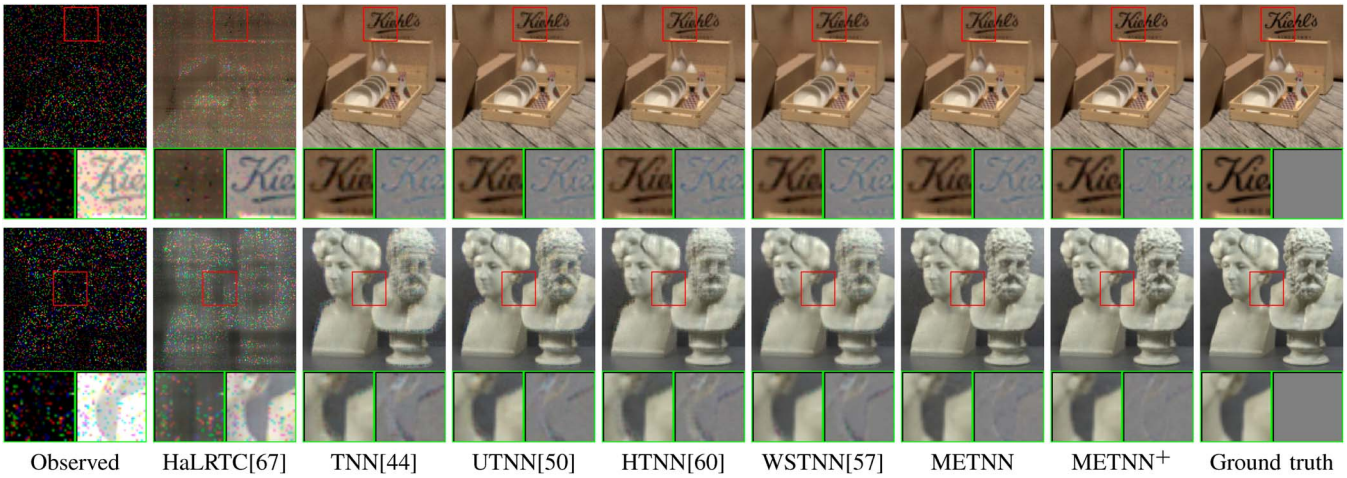


Fig. 4. The selected band of restoration results by different methods on light field images (i.e., *Dish* and *Greek*) under SR=0.1.

TABLE IV
EVALUATION INDICES OF RESTORATION RESULTS BY DIFFERENT METHODS ON LIGHT FIELD VIDEOS UNDER DIFFERENT SRS

		SR=0.05			SR=0.10			SR=0.15			SR=0.20		
Data	Model	PSNR	SSIM	MSE	PSNR	SSIM	MSE	PSNR	SSIM	MSE	PSNR	SSIM	MSE
Wonam01	HaLRTC	10.569	0.213	0.609	16.148	0.392	0.320	19.092	0.535	0.228	21.112	0.638	0.181
	TNN	18.112	0.456	0.258	25.926	0.773	0.110	28.461	0.841	0.083	30.074	0.877	0.069
	UTNN	23.395	0.680	0.145	26.847	0.798	0.099	29.500	0.864	0.074	31.564	0.902	0.058
	HTNN	18.879	0.509	0.235	28.080	0.834	0.084	30.723	0.889	0.062	32.355	0.916	0.051
	WSTNN	25.007	0.783	0.121	29.067	0.887	0.078	31.439	0.923	0.060	33.299	0.944	0.048
	METNN	27.011	0.822	0.094	30.508	0.901	0.064	32.701	0.933	0.050	34.371	**0.951**	0.042
	METNN$^+$	**27.903**	**0.844**	**0.085**	**31.272**	**0.911**	**0.058**	**33.397**	**0.939**	**0.045**	**34.620**	0.950	**0.039**
Wonam02	HaLRTC	13.070	0.290	0.557	19.380	0.441	0.264	21.996	0.538	0.194	23.655	0.611	0.160
	TNN	21.225	0.458	0.215	27.048	0.727	0.111	28.823	0.794	0.090	30.097	0.834	0.078
	UTNN	24.517	0.612	0.146	27.289	0.746	0.106	29.432	0.822	0.083	31.042	0.865	0.069
	HTNN	21.733	0.512	0.202	28.320	0.794	0.094	30.395	0.853	0.074	31.920	0.886	0.062
	WSTNN	26.102	0.730	0.122	28.993	0.847	0.088	30.931	0.895	0.070	32.499	0.923	0.059
	METNN	27.594	0.751	0.101	30.852	0.873	0.070	33.034	**0.918**	0.055	34.703	**0.941**	**0.045**
	METNN$^+$	**28.187**	**0.781**	**0.095**	**31.048**	**0.874**	**0.068**	**33.260**	0.916	**0.053**	**34.810**	0.940	**0.045**

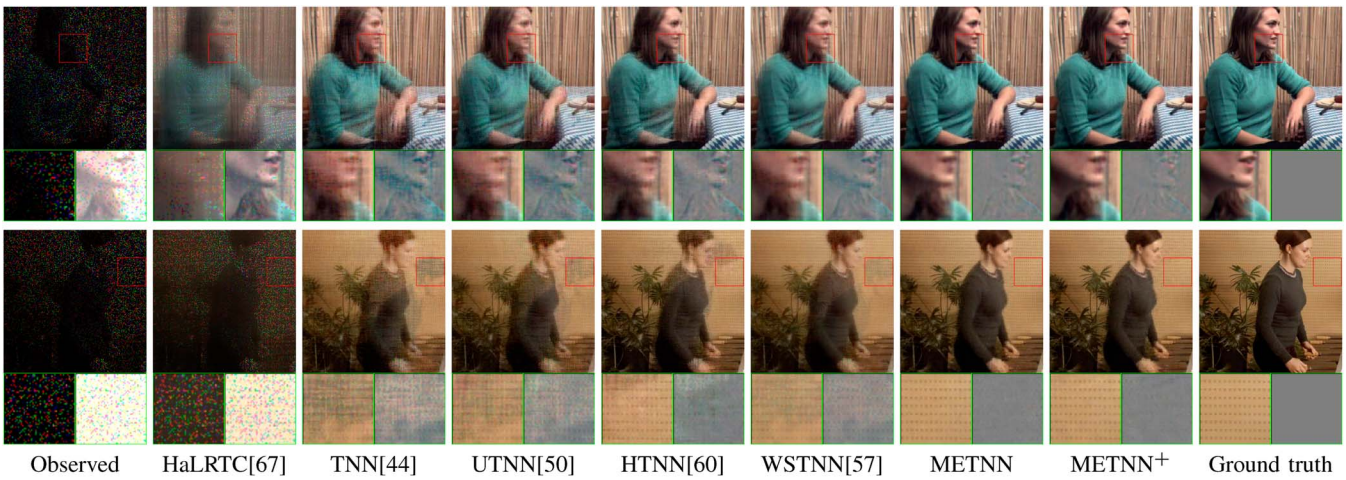
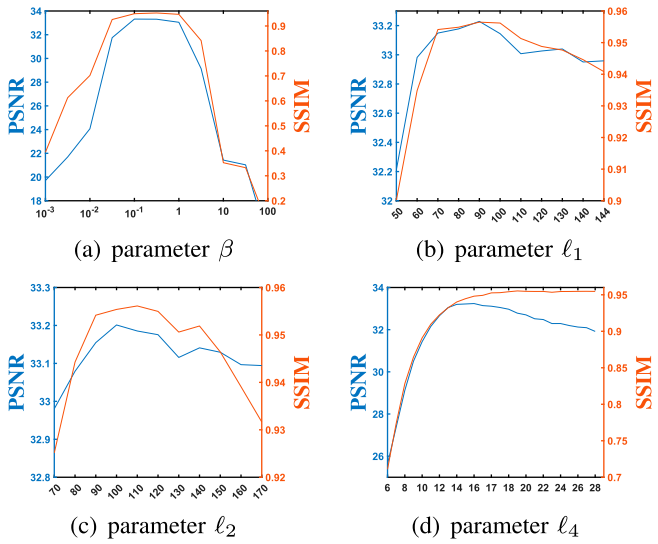


Fig. 5. The selected band of restoration results by different methods on light field videos under SR=0.1 on *Woman01* and *Woman02*.

optimal parameter β is set to 5×10^{-1} for the color video *Rhino*. Immediately after that, we discuss the influence of ℓ_1, ℓ_2, and ℓ_4. In Fig. 6(b)–6(d), we display the PSNR and SSIM values for different settings of ℓ_1, ℓ_2, and ℓ_4, where ℓ_1 is selected from $\{50, 60, \cdots, 130, 140, 144\}$, ℓ_2 is selected from $\{70, 80, \cdots, 160, 170\}$, and ℓ_4 is selected from $\{6, 7, \cdots, 27, 28\}$. In this case, the optimal parameters of ℓ_1 is 90, ℓ_2 is 100, and ℓ_4 is 15.

TABLE V
THE RESTORATION RESULTS OF DIFFERENT METHODS ON TRAFFIC DATA

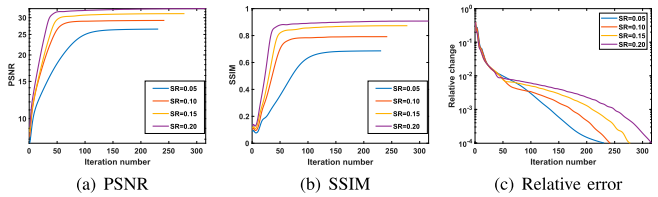
Data	Method	SR=0.05			SR=0.1			SR=0.15			SR=0.2		
		MSE	MAPE	RMSE	MSE	MAPE	RMSE	MSE	MAPE	RMSE	MSE	MAPE	RMSE
Pems-8W	Observed	0.975	95.001	0.744	0.949	89.995	0.724	0.922	84.995	0.703	0.894	79.998	0.682
	HaLRTC	0.098	9.290	0.075	0.071	5.973	0.054	0.061	4.689	0.046	0.054	3.892	0.041
	TNN	0.055	3.959	0.042	0.044	2.976	0.034	0.037	2.385	0.028	0.032	1.986	0.025
	UTNN	0.052	3.642	0.039	0.040	2.633	0.031	0.034	2.085	0.026	0.029	1.794	0.022
	HTNN	0.055	3.958	0.042	0.044	2.978	0.034	0.037	2.385	0.028	0.032	1.987	0.025
	WSTNN	0.051	3.653	0.039	0.040	2.658	0.031	0.034	2.125	0.026	0.030	1.792	0.023
	MTENN	**0.047**	3.112	**0.036**	**0.038**	2.365	**0.029**	**0.032**	1.935	**0.025**	**0.029**	1.653	**0.022**
	MTENN+	**0.047**	**3.083**	**0.036**	**0.038**	**2.333**	**0.029**	**0.032**	**1.909**	**0.025**	**0.029**	**1.628**	**0.022**
Pems-12W	Observed	0.975	95.002	0.799	0.949	90.006	0.777	0.922	85.013	0.756	0.895	80.022	0.733
	HaLRTC	0.090	7.923	0.074	0.063	4.709	0.052	0.053	3.625	0.043	0.046	2.974	0.038
	TNN	0.049	3.047	0.040	0.040	2.381	0.032	0.033	1.931	0.027	0.029	1.605	0.024
	UTNN	0.045	2.822	0.037	0.036	2.116	0.029	0.030	1.703	0.024	0.026	1.420	0.021
	HTNN	0.049	3.048	0.040	0.040	2.388	0.033	0.033	1.931	0.027	0.029	1.606	0.024
	WSTNN	0.042	2.563	0.034	0.033	1.914	0.027	0.028	1.540	0.023	**0.024**	1.298	**0.020**
	MTENN	0.040	2.284	0.033	0.032	1.745	**0.026**	**0.027**	1.447	0.023	**0.024**	1.231	**0.020**
	MTENN+	**0.039**	**2.208**	**0.032**	**0.031**	**1.704**	**0.026**	**0.027**	**1.416**	**0.022**	**0.024**	**1.206**	**0.020**



(a) parameter β (b) parameter ℓ_1

(c) parameter ℓ_2 (d) parameter ℓ_4

Fig. 6. The PSNR and SSIM values with respect to different parameter settings on color video *Flight* under SR = 0.1.



(a) PSNR (b) SSIM (c) Relative error

Fig. 7. The PSNR, SSIM, and relative error curves versus the number of iterations on color video *Rhino*.

2) Convergence Analysis: In this part, to corroborate Theorem 5, we conduct numerical experiments to analyze the convergent behavior of the proposed model on color video *Rhino*. Fig. 7 displays the PSNR, SSIM, and relative error curves versus the number of iterations at different SRs. As the number of iterations increases, the PSNR and SSIM values stabilize, and the relative error gradually decreases to 10^{-4}. This further confirms that the proposed ADMM algorithm is convergent.

VII. CONCLUSION

In this paper, we propose a novel elt-product as a generalization of third-order t-product for dth-order tensor, where its basic element is a $(d-2)$th-order tensor instead of a vector. The elt-product can avoid the unfolding operation and explore more complex interactions via higher-order convolution instead of limiting the first-order convolution, which better explores the underlying structure of high-order tensor data. Based on the elt-product, we establish the corresponding LRTC model and prove the exact recovery guarantee of the proposed LRTC model. To harness the resulting nonconvex optimization problem, we apply an ADMM algorithm and prove the convergence of the developed algorithm. Extensive experimental results on the simulated and real-world data (color videos, light-field images, and light-field videos) demonstrate the superiority of the proposed models against the state-of-the-art baseline models.

REFERENCES

[1] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE Trans. Signal Process.*, vol. 65, no. 13, pp. 3551–3582, Jul. 2017.

[2] A. Cichocki et al., "Tensor decompositions for signal processing applications: From two-way to multiway component analysis," *IEEE Signal Process. Mag.*, vol. 32, no. 2, pp. 145–163, Mar. 2015.

[3] P. M. Hoang, H. D. Tuan, T. T. Son, H. V. Poor, and L. Hanzo, "Learning unbalanced and sparse low-order tensors," *IEEE Trans. Signal Process.*, vol. 70, pp. 5624–5638, 2022.

[4] X. Fu, N. Vervliet, L. De Lathauwer, K. Huang, and N. Gillis, "Computing large-scale matrix and tensor decomposition with structured factors: A unified nonconvex optimization perspective," *IEEE Signal Process. Mag.*, vol. 37, no. 5, pp. 78–94, Sep. 2020.

[5] J. Miao and K. I. Kou, "Quaternion-based bilinear factor matrix norm minimization for color image inpainting," *IEEE Trans. Signal Process.*, vol. 68, pp. 5617–5631, 2020.

[6] X. Fu, S. Ibrahim, H.-T. Wai, C. Gao, and K. Huang, "Block-randomized stochastic proximal gradient for low-rank tensor factorization," *IEEE Trans. Signal Process.*, vol. 68, pp. 2170–2185, 2020.

[7] L. Yang, Z.-H. Huang, and X. Shi, "A fixed point iterative method for low n-rank tensor pursuit," *IEEE Trans. Signal Process.*, vol. 61, no. 11, pp. 2952–2962, Jun. 2013.

[8] X. Gong, W. Chen, and J. Chen, "A low-rank tensor dictionary learning method for hyperspectral image denoising," *IEEE Trans. Signal Process.*, vol. 68, pp. 1168–1180, 2020.

[9] C. Prévost, K. Usevich, P. Comon, and D. Brie, "Hyperspectral super-resolution with coupled tucker approximation: Recoverability and SVD-based algorithms," *IEEE Trans. Signal Process.*, vol. 68, pp. 931–946, 2020.

[10] Z. Jia, Q. Jin, M. K. Ng, and X.-L. Zhao, "Non-local robust quaternion matrix completion for large-scale color image and video inpainting," *IEEE Trans. Image Process.*, vol. 31, pp. 3868–3883, 2022.

[11] Q.-T. Truong et al., "Marine video kit: A new marine video dataset for content-based analysis and retrieval," in *Proc. Int. Conf. Multimedia Model.*, New York, NY, USA: Springer-Verlag, 2023, pp. 539–550.

[12] M. H. Kamal, B. Heshmat, R. Raskar, P. Vandergheynst, and G. Wetzstein, "Tensor low-rank and sparse light field photography," in *Computer Vision and Image Understanding*, vol. 145, Amsterdam, Netherlands: Elsevier, 2016, pp. 172–181.

[13] K. Honauer, O. Johannsen, D. Kondermann, and B. Goldluecke, "A dataset and evaluation methodology for depth estimation on 4d light fields," in *Proc. Asian Conf. Comput. Vis.*, New York, NY, USA: Springer-Verlag, 2016.

[14] X. Chen, M. Lei, N. Saunier, and L. Sun, "Low-rank autoregressive tensor completion for spatiotemporal traffic data imputation," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 12301–12310, Aug. 2022.

[15] X. Chen and L. Sun, "Bayesian temporal factorization for multidimensional time series prediction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 4659–4673, Sep. 2022.

[16] W.-J. Zheng, X.-L. Zhao, Y.-B. Zheng, J. Lin, L. Zhuang, and T.-Z. Huang, "Spatial-spectral-temporal connective tensor network decomposition for thick cloud removal," *ISPRS J. Photogrammetry Remote Sens.*, vol. 199, pp. 182–194, May 2023.

[17] L. Li, W. Li, Q. Du, and R. Tao, "Low-rank and sparse decomposition with mixture of Gaussian for hyperspectral anomaly detection," *IEEE Trans. Cybern.*, vol. 51, no. 9, pp. 4363–4372, Sep. 2021.

[18] H. Zeng, S. Huang, Y. Chen, S. Liu, H. Q. Luong, and W. Philips, "Tensor completion using bilayer multimode low-rank prior and total variation," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, May 17, 2023, doi:110.1109/TNNLS.2023.3266841.

[19] Z. Long, C. Zhu, J. Liu, P. Comon, and Y. Liu, "Trainable subspaces for low rank tensor completion: Model and analysis," *IEEE Trans. Signal Process.*, vol. 70, pp. 2502–2517, 2022.

[20] 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.*, vol. 43, no. 10, pp. 3492–3507, Oct. 2021.

[21] Y. Yang, L. Han, Y. Liu, J. Zhu, and H. Yan, "A novel regularized model for third-order tensor completion," *IEEE Trans. Signal Process.*, vol. 69, pp. 3473–3483, 2021.

[22] T. Yokota, Q. Zhao, and A. Cichocki, "Smooth PARAFAC decomposition for tensor completion," *IEEE Trans. Signal Process.*, vol. 64, no. 20, pp. 5423–5436, Oct. 2016.

[23] G. Zhang, X. Fu, J. Wang, X.-L. Zhao, and M. Hong, "Spectrum cartography via coupled block-term tensor decomposition," *IEEE Trans. Signal Process.*, vol. 68, pp. 3660–3675, 2020.

[24] M. Yang, Q. Luo, W. Li, and M. Xiao, "3-D array image data completion by tensor decomposition and nonconvex regularization approach," *IEEE Trans. Signal Process.*, vol. 70, pp. 4291–4304, 2022.

[25] J.-H. Yang, C. Chen, H.-N. Dai, M. Ding, Z.-B. Wu, and Z. Zheng, "Robust corrupted data recovery and clustering via generalized transformed tensor low-rank representation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 7, pp. 8839–8853, Jul. 2024.

[26] X.-L. Zhao, J.-H. Yang, T.-H. Ma, T.-X. Jiang, M. K. Ng, and T.-Z. Huang, "Tensor completion via complementary global, local, and nonlocal priors," *IEEE Trans. Image Process.*, vol. 31, pp. 984–999, 2021.

[27] K. Huang, N. D. Sidiropoulos, and A. P. Liavas, "A flexible and efficient algorithmic framework for constrained matrix and tensor factorization," *IEEE Trans. Signal Process.*, vol. 64, no. 19, pp. 5052–5065, Oct. 2016.

[28] L. Cheng, Y.-C. Wu, and H. V. Poor, "Scaling probabilistic tensor canonical polyadic decomposition to massive data," *IEEE Trans. Signal Process.*, vol. 66, no. 21, pp. 5534–5548, Nov. 2018.

[29] O. A. Malik, S. Ubaru, L. Horesh, M. E. Kilmer, and H. Avron, "Dynamic graph convolutional networks using the tensor M-product," in *Proc. SIAM Int. Conf. Data Mining (SDM),* Philadelphia, PA, USA: SIAM, 2021, pp. 729–737.

[30] F. L. Hitchcock, "The expression of a tensor or a polyadic as a sum of products," *J. Math. Phys.*, vol. 6, nos. 1–4, pp. 164–189, 1927.

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

[32] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.

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

[34] 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.

[35] Q. Zhao, G. Zhou, S. Xie, L. Zhang, and A. Cichocki, "Tensor ring decomposition," 2016, *arXiv:1606.05535*.

[36] X. P. Li and H. C. So, "Robust low-rank tensor completion based on tensor ring rank via p, ϵ-norm," *IEEE Trans. Signal Process.*, vol. 69, pp. 3685–3698, 2021.

[37] Y. Qiu, G. Zhou, Q. Zhao, and S. Xie, "Noisy tensor completion via low-rank tensor ring," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 1, pp. 1127–1141, Jan. 2024.

[38] 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.*, 2019, vol. 33, no. 1, pp. 9151–9158.

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

[40] M. E. Kilmer, L. Horesh, H. Avron, and E. Newman, "Tensor-tensor algebra for optimal representation and compression of multiway data," *Proc. Nat. Acad. Sci.*, vol. 118, no. 28, 2021, Art. no. e2015851118.

[41] Y.-B. Zheng, T.-Z. Huang, X.-L. Zhao, Q. Zhao, and T.-X. Jiang, "Fully-connected tensor network decomposition and its application to higher-order tensor completion," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 12, pp. 11071–11078.

[42] 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.

[43] 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.*, 2014, pp. 3842–3849.

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

[45] 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.

[46] K. Gilman, D. A. Tarzanagh, and L. Balzano, "Grassmannian optimization for online tensor completion and tracking with the t-SVD," *IEEE Trans. Signal Process.*, vol. 70, pp. 2152–2167, 2022.

[47] M. E. Kilmer, L. Horesh, H. Avron, and E. Newman, "Don't matricize, tensorize: Tensor-tensor products for optimal representation and compression," in *Proc. XXI Householder Symp. Numer. Linear Algebra*, 2020, p. 229.

[48] 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.*, 2019, pp. 5996–6004.

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

[50] 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, 2020, Art. no. e2299.

[51] 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.

[52] H. Kong, C. Lu, and Z. Lin, "Tensor Q-rank: New data dependent definition of tensor rank," *Mach. Learn.*, vol. 110, no. 7, pp. 1867–1900, 2021.

[53] T.-X. Jiang, X.-L. Zhao, H. Zhang, and M. K. Ng, "Dictionary learning with low-rank coding coefficients for tensor completion," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 2, pp. 932–946, Feb. 2023.

[54] S. Liu, J. Leng, X.-L. Zhao, H. Zeng, Y. Wang, and J.-H. Yang, "Learnable spatial-spectral transform-based tensor nuclear norm for

multi-dimensional visual data recovery," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 5, pp. 3633–3646, May 2024.
[55] B.-Z. Li, X.-L. Zhao, T.-Y. Ji, X.-J. Zhang, and T.-Z. Huang, "Nonlinear transform induced tensor nuclear norm for tensor completion," *J. Sci. Comput.*, vol. 92, no. 3, 2022, Art. no. 83.
[56] J.-L. Wang, T.-Z. Huang, X.-L. Zhao, Y.-S. Luo, and T.-X. Jiang, "CoNoT: Coupled nonlinear transform-based low-rank tensor representation for multidimensional image completion," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 7, pp. 8969–8983, Jul. 2024.
[57] 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, 2020.
[58] A. Wang, Q. Zhao, Z. Jin, C. Li, and G. Zhou, "Robust tensor decomposition via orientation invariant tubal nuclear norms," *Sci. China Technol. Sci.*, vol. 65, no. 6, pp. 1300–1317, 2022.
[59] 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.
[60] W. Qin, H. Wang, F. Zhang, J. Wang, X. Luo, and T. Huang, "Low-rank high-order tensor completion with applications in visual data," *IEEE Trans. Image Process.*, vol. 31, pp. 2433–2448, 2022.
[61] Y. Wang and Y. Yang, "Hot-SVD: Higher order t-singular value decomposition for tensors based on tensor–tensor product," *Comput. Appl. Math.*, vol. 41, no. 8, 2022, Art. no. 394.
[62] Y.-B. Zheng, T.-Z. Huang, X.-L. Zhao, T.-X. Jiang, T.-H. Ma, and T.-Y. Ji, "Mixed noise removal in hyperspectral image via low-fibered-rank regularization," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 1, pp. 734–749, Jan. 2020.
[63] S. Boyd, N. Parikh, and E. Chu, *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*. Boston, MA, USA: Now, 2011.
[64] 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.
[65] P. H. Schönemann, "A generalized solution of the orthogonal procrustes problem," *Psychometrika*, vol. 31, no. 1, pp. 1–10, 1966.
[66] Y. Wang, W. Yin, and J. Zeng, "Global convergence of admm in nonconvex nonsmooth optimization," *J. Sci. Comput.*, vol. 78, pp. 29–63, Jun. 2019.
[67] 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.
[68] L. Dabala et al., "Efficient multi-image correspondences for on-line light field video processing," *Comput. Graph. Forum*, vol. 35, no. 7, pp. 401–410, Oct. 2016.
[69] X. Chen, Y. Chen, N. Saunier, and L. Sun, "Scalable low-rank tensor learning for spatiotemporal traffic data imputation," *Transp. Res. Part C: Emerg. Technol.*, vol. 129, 2021, Art. no. 103226.

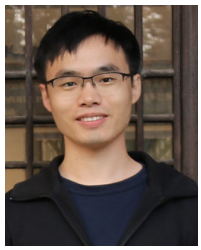
Sheng Liu received the B.S. degree in information and computer science from the Northwest A&F University, Yangling, China, in 2022. He is currently working toward the Ph.D. degree with the School of Mathematical Sciences, University of Electronic Science and Technology of China (UESTC), Chengdu, China. His research interest includes tensor modeling and algorithms for high-order data recovery.

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 worked as a Postdoc with Prof. M. Ng at Hong Kong Baptist University from 2013 to 2014. He worked as a Visiting Scholar with Prof. J. Bioucas Dias at the University of Lisbon from 2016 to 2017. He is currently a Professor with the School of Mathematical Sciences, UESTC. His research interests include image processing, machine learning, and scientific computing. For more information, see https://zhaoxile.github.io/.

Jinsong Leng received the B.E. and M.S. degrees in applied mathematics from the Department of Mathematics, Sichuan Normal University, Chengdu, China, in 1990 and 1993, respectively, and the Ph.D. degree in computational mathematics from the Department of Mathematics, Xi'an Jiaotong University, Xi'an, China, in 2006. From 2010 to 2011, he was a Visiting Scholar with the Department of Mathematics, University of Central Florida, Orlando, FL, USA. He is currently a Professor and a Doctoral Tutor with the School of Mathematical Sciences, UESTC, Chengdu, China. His research interests include wavelet analysis with its applications, frame theory with its application in signal processing, and signal and image processing.

Ben-Zheng Li (Graduate Student Member, IEEE) received the B.S. degree from the School of Science, East China University of Technology, Nanchang, China, in 2019. He is currently working toward the Ph.D. degree with the School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, China. His research interests include tensor modeling and algorithm for high-dimensional data recovery. For more information, see https://benzhengli.github.io/.

Jing-Hua Yang (Member, IEEE) received the Ph.D. degree from Macau University of Science and Technology, Macau, China, in 2023. She is currently an Assistant Professor with the School of Information Science and Technology, Southwest Jiaotong University. Her research interests include data mining, image processing, and deep learning.



Xinyu Chen received the Ph.D. degree from the University of Montréal, Montreal, QC, Canada. He is currently a Postdoctoral Associate with the Massachusetts Institute of Technology, Cambridge, MA, USA. His research interests include machine learning, spatiotemporal data modeling, intelligent transportation systems, and urban science.