DELTA: Deep Low-Rank Tensor Representation for Multi-Dimensional Data Recovery

Guo-Wei Yang, Liqiao Yang, Tai-Xiang Jiang, *Member, IEEE*, Guisong Liu, *Member, IEEE*, and Michael K. Ng

***Abstract*—Low-rank tensor recovery methods within the tensor singular value decomposition (t-SVD) framework have demonstrated considerable success by leveraging the inherent low-dimensional structures of multi-dimensional data. However, previous approaches in this framework often rely on linear transforms or, in some cases, nonlinear transforms constructed with fully connected networks (FCNs). These methods typically promote a global low-rank structure, which may not fully exploit the nature of multiple subspaces in real-world data. In this work, we propose a nonlinear transform to capture long-range dependencies and diverse patterns across multiple subspaces of the data within the t-SVD framework. This approach provides a richer and more nuanced representation compared to the localized processing typically seen in FCN-based transforms. In the transform domain, we construct a low-rank self-representation layer that fully exploits the multi-subspace structure inherent in tensor data. Instead of merely enforcing overall low-rankness, our method minimizes the nuclear norm of a self-representation tensor, allowing for a more precise and joint characterization of multiple subspaces. This results in a more accurate representation of the data's intrinsic low-dimensional structures, leading to superior recovery performance. This new framework, termed the DEep Low-rank Tensor representAtion (DELTA), is evaluated across several typical multi-dimensional data recovery applications, including tensor completion, robust tensor completion, and spectral snapshot imaging. Experiments on various real-world multi-dimensional data illustrate the superior performance of our DELTA.**

***Index Terms*—Tensor singular value decomposition, low-rank tensor representation, deep nonlinear transform, multi-dimensional data recovery.**

Received 10 October 2024; revised 12 October 2025; accepted 4 November 2025. Date of publication 7 November 2025; date of current version 4 February 2026. This work was supported in part by the Sichuan Science and Technology Program under Grant 2024ZYD0147, Grant 2024NSFSC1452, and Grant 2024NSFSC0038; in part by the Natural Science Foundation of Xinjiang Uygur Autonomous Region under Grant 2024D01A18; in part by GDSTC: Guangdong and Hong Kong Universities "1+1+1" Joint Research Collaboration Scheme under Grant UICR0800008-24, in part by National Key Research and Development Program of China under Grant 2024YFE0202900, in part by RGC GRF under Grant 12300125, and in part by Joint NSFC and RGC under Grant N-HKU769/21 Recommended for acceptance by Y. Yu. *(Corresponding author: Tai-Xiang Jiang.)*

Guo-Wei Yang, Liqiao Yang, Tai-Xiang Jiang, and Guisong Liu are with the School of Computing and Artificial Intelligence, Southwestern University of Finance and Economics, Chengdu 610074, China, and also with the Kash Institute of Electronics and Information Industry, Kashgar 844000, China (e-mail: gwyang1013@163.com; liqiaoyoung@163.com; taixiangjiang@gmail.com; gliu@swufe.edu.cn).

Michael K. Ng is with the Department of Mathematics, Hong Kong Baptist University, Kowloon Tong Hong Kong (e-mail: michael-ng@hkbu.edu.hk).

Digital Object Identifier 10.1109/TPAMI.2025.3630339

I. Introduction

RECOVERING multi-dimensional data from degraded observations or compressed measurements is an essential task for many real-world applications, e.g., computer vision [1], [2], [3], compressive sensing magnetic resonance imaging (MRI) [4], hyperspectral analysis [5], and intelligent transportation [6]. Generally, it is necessary to fully explore the prior knowledge to tackle this typical inverse problem. As multi-dimensional data are always internally correlated in real-world scenarios and naturally in the tensor format, the tensor low-rank prior (i.e., the low-dimensional structure) is widely utilized (see [7], [8], [9], [10]).

Although the tensor format has shown superiority over the matrix format in characterizing the inner structure of multi-dimensional data, the definition of the tensor rank is not unique and varies depending on tensor decomposition schemes, widely used examples being the CANDECOMP/PARAFAC decomposition [11], [12], Tucker decomposition [13], tensor train decomposition [14], tensor ring decomposition [15], and tensor singular value decomposition (t-SVD) [16], [17], [18]. This work focuses on the t-SVD framework, in which the tensor-tensor product (t-prod) is well defined with a tensor algebraic framework that is highly analogous to the matrix case. Meanwhile, it has been shown to be capable of capturing the spatial-shifting correlation and characterizing the intrinsic structure of a third-order tensor. Nonetheless, the t-SVD has been effectively extended for higher-order tensors [19], [20]. The superiority of tensor modeling capacity based on the t-SVD framework has been theoretically analyzed [21], [22] and empirically validated in a wide range of real-world applications.

In t-SVD, equipped with the t-prod, the tensor tubal-rank [8], [23] can be derived. Similarly to the matrix case, one can minimize the tubal nuclear norm (TNN) to fully enhance the tensor low-rankness of the underlying data. Benefiting from the particular design of the t-prod, minimizing the TNN only requires three steps: (i) applying the discrete Fourier transformation (DFT) along the third mode, (ii) minimizing the matrix nuclear norm, which can be readily accomplished by the singular value thresholding operation [24], of the frontal slices, and (iii) applying the inverse DFT along the third mode. More mathematical details will be provided in Section II. Moreover, [25] noted that the t-prod can be defined with any linear invertible transform. Subsequently, numerous efforts have been made to find substitutes that are more suitable for multi-dimensional

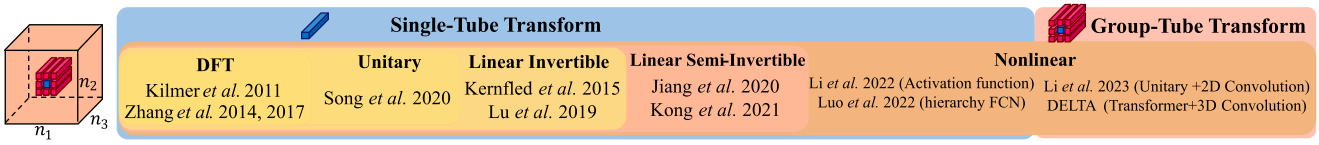

Fig. 1. Representative works on transforms within the t-SVD framework.

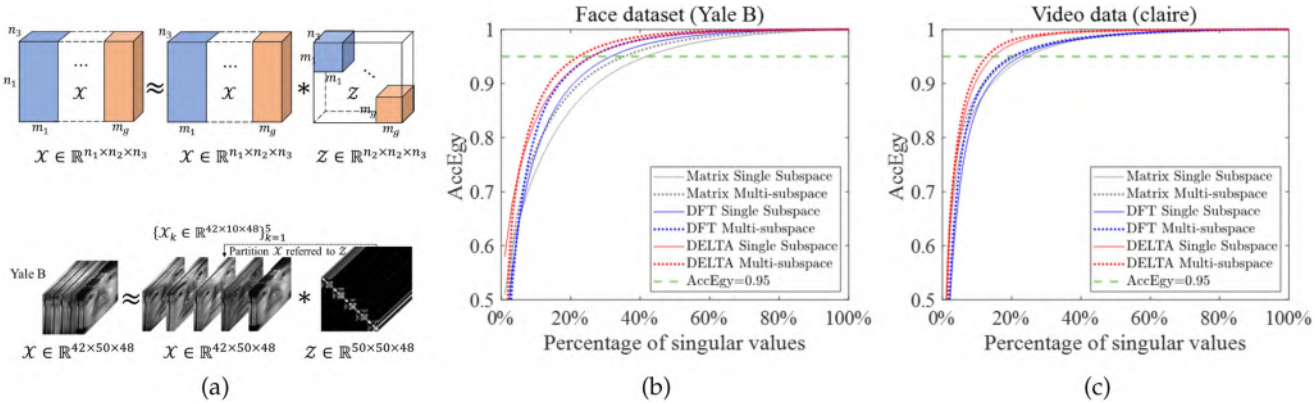

Fig. 2. **(a)** Illustration of tensor self-representation (top) and a visualization of a union-of-subspaces partition on *Yale B* (bottom). The partitions $\{\mathcal{X}_k\}_{k=1}^5$ of the whole face data tensor $\mathcal{X}$ can be obtained directly from the block-diagonal structure of the representation coefficient tensor $\mathcal{Z}$. We note that our method does *NOT* require this explicit partition, and it is used herein *only* for visualization and quantitative analysis. **(b, c)** Accumulated energy (AccEgy) curves of different approaches on two datasets. For singular values $\{\sigma_i\}$ (sorted non-increasing), $\text{AccEgy}(k) = \sum_{i=1}^{k} \sigma_i^2 \Big/ \sum_j \sigma_j^2$, plotted versus the percentage of singular values. Matrix case (black): "*Single Subspace*" indicates the SVD of the mode-2 unfolding of the whole data $\mathcal{X}$, whereas "*Multi-subspace*" first computes SVDs on the mode-2 unfolding of each partition (subtensor) and then concatenates and sorts all singular values to form one AccEgy curve. Tensor case (blue/red): After mapping to the transform domain, linear DFT for blue and our nonlinear transform for red, we compute SVD per frontal slice. For "*Single Subspace*" we form an AccEgy curve for each slice and plot the mean across all n_3 slices. For "*Multi-subspace*" (dashed blue and red), at each slice index we take the slice from every partition, compute their SVDs, concatenate and sort the singular values for that slice, form one AccEgy curve, and then plot the mean over slices. The faster rise of dashed lines (multi-subspace) over solid lines (single subspace) confirms the benefit of multi-subspace modeling. Furthermore, the hierarchy of the curves shows that tensor-native models (blue, red) outperform matrix-based ones (black), and that our nonlinear DELTA (red) provides the most compact representation compared to the linear DFT-based model (blue).

data [9], [26], [27], [28], [29]. To capture the implicit low-rank nature of the data, nonlinear transforms [10], [30], [31], [32] have recently been considered by incorporating different deep neural networks. The evolution of transforms (including the proposed one in this paper) in the t-SVD framework is summarized in Fig. 1.

Previous t-SVD based recovery methods (whether using fixed transforms or learned transforms via fully-connected networks [30]) typically impose a global low-rank constraint on the entire tensor. This can be viewed as finding a single low-dimensional subspace (or manifold) where all samples reside. While this is reasonable due to the inner correlations in real-world data (e.g., temporal continuity in videos or spectral correlation in hyperspectral images), these data often lie in a **union of multiple fine subspaces** rather than a single global one. A representative example is face data. Face images from a dataset share similar spatial geometries and thus lie in a subspace whose dimension is much smaller than the ambient image space [33], [34]. However, face images of *different individuals* will lie in their own distinct, smaller subspaces [35]. The challenge of modeling multiple subspaces has been studied in the matrix setting via low-rank representation (LRR) [36], and later extended to tensors through tensor low-rank representation (TLRR) within the t-SVD framework [22]. As shown in Fig. 2(a), after appropriate permutations of lateral slices in the data tensor, enforcing low-rankness on a self-representation coefficient tensor induces a block-diagonal structure, corresponding to the multiple subspaces.

To quantitatively illustrate the benefits of modeling these fine subspaces, we analyze the *Yale B* face dataset and the *claire* video sequence. For this analysis only,[1] we partition the data into subsets of lateral slices based on the block-diagonal self-representation structure. The plots of the accumulation energy (AccEgy) in Fig. 2(b) and (c) reveal three consistent findings. First, multi-subspace cases (dashed lines) exhibit significantly faster AccEgy growth than **single-subspace** counterparts (solid lines), confirming that union-of-subspaces modeling more compact representations. Second, tensor-native cases (blue and red) achieve faster singular value decay than the matrix case (black), highlighting the benefit of preserving multi-way tensor structures. Third, within tensor methods, the proposed DELTA with nonlinear transform (red) yields the most compact representation, outperforming the linear DFT-based tensor case (blue). These findings provide strong empirical motivation for

[1] We note that this explicit partition is not required in LRR, TLRR, or our proposed method. It is used here purely for visualization and quantitative analysis.

a framework that combines multi-subspace modeling, tensor-native representations, and nonlinear transforms.

The main challenge, therefore, is to develop a unified framework that simultaneously models the tensor data, captures complex nonlinear dependencies, and exploits the multi-subspace structure of multi-dimensional data for recovery tasks. In the literature, deep matrix methods like [37], [38] handle nonlinearity and multiple subspaces in a matrix setting but do not leverage the rich structural information preserved in multiple tensor subspaces. Tensor-based methods [22], [39] leverage tensor algebra for multi-subspace modeling but lack the capacity for nonlinear feature extraction critical for many real-world data.

To address this gap, we develop a novel DEep Low-rank Tensor representAtion (DELTA) framework in this paper. Our DELTA operates within the tensor algebraic skeleton of the t-SVD. We propose and develop an effective nonlinear transform to deal with both local and long-range nonlinear patterns. We construct a low-rank self-representation tensor (multiple subspaces of low-rank) that can be obtained by minimizing its nuclear norm in the underlying tensor data. The main contributions of this paper are given as follows:

- We propose a novel group-tube nonlinear transform combining a 1D spectral/temporal global transform with a 2D spatial transform. The 1D transform utilizes the encoder architecture of the Transformer model [40], incorporating multi-head attention to accurately capture complex nonlinear characteristics and intricate intrinsic features across diverse data types, particularly as the properties of the third dimension vary significantly. To further exploit local homogeneity, we leverage the 3D convolution [41] with $3 \times 3 \times 1$ kernels as the spatial local transform. Our proposed nonlinear transform can effectively capture the correlations among global and neighboring tubes within the multi-dimensional data, carving out data features at a more granular level and thereby yielding a more compact representation.
- In the deep transform domain, a self-representation tensor is established, seamlessly embedded into the nonlinear transform-based t-SVD framework. This allows for the efficient characterization of the subspace distribution by simultaneously identifying the low-rank representation. Thus, the multi-subspace structure of the multi-dimensional data can be effectively captured by enhancing its low-rankness in the deep transform domain.
- We apply our deep low-rank tensor representation, which incorporates the newly proposed deep nonlinear transform and deep self-representation, for multi-dimensional data recovery. All the learnable parameters, i.e., the network parameters in the nonlinear transform and deep representation coefficients, can be inferred directly from the degraded data itself in a zero-shot learning manner, using only the observed data. Extensive numerical experiments on various real-world multi-dimensional data for different recovery tasks demonstrate that our method outperforms state-of-the-art methods. Furthermore, discussions show that our method is scalable to higher-order tensors.

The rest of this article is organized as follows. Section II gives the basic notations and preliminaries. Section III presents the proposed DEep Low-rank Tensor RepresentAtion (DELTA). We show experimental results and discussions in Section IV. Finally, conclusions are drawn in Section V.

II. Preliminaries

We use lowercase letters, e.g., x, to denote scalars, boldface lowercase letters, e.g., $\mathbf{x}$, for vectors, boldface uppercase letters, e.g., $\mathbf{X}$, for matrices, and boldface calligraphic letters, e.g., $\mathcal{X}$, for tensors. As for third-order tensors, we use $\mathcal{X}^{(k)}$ to denote the k-th frontal slice, i.e., $\mathcal{X}^{(k)} = \mathcal{X}(:,:,k)$. A frequently used matricization operation is the mode-3 unfolding, which maps the (i,j,k)-th element of a tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ to the matrix's (k,l)-th element satisfying $l = (j-1)n_1 + i$. We use $\texttt{unfold}_3$ and $\texttt{fold}_3$ to denote it and its inverse operation, respectively. The mode-3 tensor-matrix product of a tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ with a matrix $\mathbf{A} \in \mathbb{R}^{m \times n_3}$ is denoted by $\mathcal{X} \times_3 \mathbf{A}$ and is of size $n_1 \times n_2 \times m$. Element-wise, we have $(\mathcal{X} \times_3 \mathbf{A})_{ijk} = \sum_{n=1}^{n_3} x_{ijn} a_{kn}$. The mode-3 tensor-matrix product can also be equivalently expressed in terms of the mode-3 unfolding as $\mathcal{Y} = \mathcal{X} \times_3 \mathbf{A} \Leftrightarrow \texttt{unfold}_3(\mathcal{Y}) = \mathbf{A} \cdot \texttt{unfold}_3(\mathcal{X})$, where $\cdot$ is the matrix product. More details of tensor basics can be found in [42].

Definition 2.1 (Frontal-slice-wise product [25]): Given two third-order tensors $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{Y} \in \mathbb{R}^{n_2 \times n_4 \times n_3}$, the frontal-slice-wise product $\mathcal{X} \odot \mathcal{Y}$ is defined by

$$(\mathcal{X} \odot \mathcal{Y})^{(k)} = \mathcal{X}^{(k)} \mathcal{Y}^{(k)}, \tag{1}$$

for $k \in \{1, 2, \ldots, n_3\}$. The symbol $\odot$ is also referred to as the face-wise product in [25].

Next, we restate definitions of the linear transform-based t-SVD.

Definition 2.2 (Linear transform based t-prod [25]): Let $\mathbf{L} \in \mathbb{C}^{n_3 \times n_3}$ be a linear invertible transform matrix. The linear invertible transform $\mathbf{L}$ based tensor-tensor product (t-prod) between two tensors $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{B} \in \mathbb{R}^{n_2 \times n_4 \times n_3}$ is denoted as $\mathcal{C} \in \mathbb{R}^{n_1 \times n_4 \times n_3} = \mathcal{A} *_{\mathbf{L}} \mathcal{B}$, and is defined by

$$\mathcal{C} = \mathcal{A} *_{\mathbf{L}} \mathcal{B} = ((\mathcal{A} \times_3 \mathbf{L}) \odot (\mathcal{B} \times_3 \mathbf{L})) \times_3 \mathbf{L}^{-1}. \tag{2}$$

Definition 2.3 (Tensor conjugate transpose [25]): Given a linear invertible transform matrix $\mathbf{L} \in \mathbb{C}^{n_3 \times n_3}$, the tensor conjugate transpose of $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, denoted as $\mathcal{A}^H \in \mathbb{R}^{n_2 \times n_1 \times n_3}$, satisfies $(\mathcal{A} \times_3 \mathbf{L})^{(k)} = ((\mathcal{A}^H \times_3 \mathbf{L})^{(k)})^H$ for $k = 1, 2, \ldots, n_3$.

Definition 2.4 (Orthogonal tensor [25]): A tensor $\mathcal{O}$ is called an orthogonal tensor if it satisfies $\mathcal{O} *_{\mathbf{L}} \mathcal{O}^H = \mathcal{I}$, where $\mathcal{I}$ is the identity tensor satisfying that $(\mathcal{I} \times_3 \mathbf{L})^{(k)}$ is an identity matrix for $k = 1, 2, \ldots, n_3$.

Definition 2.5 (f-diagonal tensor [17]): A tensor $\mathcal{A}$ is called f-diagonal tensor if each frontal slice $\mathcal{A}^{(k)}$ is a diagonal matrix.

Definition 2.6 (Linear transform based t-SVD [25]): For $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the linear transform based t-SVD of $\mathcal{A}$ is given by

$$\mathcal{A} = \mathcal{U} *_{\mathbf{L}} \mathcal{S} *_{\mathbf{L}} \mathcal{V}^H, \tag{3}$$

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

When $\mathbf{L}$ is the discrete Fourier transform (DFT) matrix, the definition of the t-prod and t-SVD reduces to its original version proposed in [17]. With this foundational definition, the algebraic framework of third-order tensors can be well derived, and it is analogous to the matrix case.

Definition 2.7 (Tensor tubal-rank [43]): The tensor tubal-rank of a tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, denoted as $\text{rank}_t(\mathcal{A})$, is defined as the number of non-zero singular tubes in $\mathcal{S}$, where $\mathcal{S}$ is from the linear transform based t-SVD of $\mathcal{A}$: $\mathcal{A} = \mathcal{U} *_{\mathbf{L}} \mathcal{S} *_{\mathbf{L}} \mathcal{V}^H$. Formally, we can write

$$\text{rank}_{\text{t}}(\mathcal{A}) = \#\{i, \mathcal{S}(i, i, :) \neq 0\}. \tag{4}$$

Alternatively, in the transform domain, $\text{rank}_{\text{t}}(\mathcal{A}) = \max_k \text{rank}((\mathcal{A} \times_3 \mathbf{L})^{(k)})$.

Definition 2.8 (Tubal nuclear norm (TNN) [8]): The tensor nuclear norm of a tensor $\mathcal{A} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$, denoted as $\|\mathcal{A}\|_{\text{TNN}}$, is defined as the sum of singular values of all the frontal slices of $\mathcal{A}$, i.e.,

$$\|\mathcal{A}\|_{\text{TNN}} = \sum_{k=1}^{n_3} \left\| (\mathcal{A} \times_3 \mathbf{F})^{(k)} \right\|_*, \tag{5}$$

where $\mathbf{F} \in \mathbb{C}^{n_3 \times n_3}$ is a discrete Fourier transform (DFT) matrix.

Definition 2.9 (Linear transform based TNN [25]): The transform based tubal nuclear norm of a tensor $\mathcal{A} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$, denoted as $\|\mathcal{A}\|_{\text{L}-\text{TNN}}$, is defined as the sum of singular values of all the frontal slices of $\mathcal{A}$ in the transform domain, i.e.,

$$\|\mathcal{A}\|_{\text{L}-\text{TNN}} = \sum_{k=1}^{n_3} \left\| (\mathcal{A} \times_3 \mathbf{L})^{(k)} \right\|_*, \tag{6}$$

where $\mathbf{L} \in \mathbb{C}^{n_3 \times n_3}$, acting as the transform, is a linear invertible matrix.

From Definition 2.9, we can see that the TNN, which is widely used as the convex surrogate of the tensor tubal-rank defined within t-SVD, can be directly computed in the transform domain. This also enables a flexible utilization of semi- or non-invertible transforms [28], [30], [44] wherein the low-rank promotion can be well established in the transform domain.

A *tensor space* is defined as a set $\mathbb{S} = \{\mathcal{S} \in \mathbb{R}^{n_1 \times 1 \times n_3}\}$, which is closed under finite tensor addition and scalar multiplication. A set of tensors $\{\mathcal{D}_{(1)}, \ldots, \mathcal{D}_{(p)}\} \subseteq \mathbb{S}$, where $\mathcal{D}_{(t)}$ is the t-th lateral slice of $\mathcal{D} \in \mathbb{R}^{n_1 \times p \times n_3}$, is *linearly independent* if there is no non-zero tensor $\mathcal{C} \in \mathbb{R}^{p \times 1 \times n_3}$ satisfying $\mathcal{D} *_{\mathbf{L}} \mathcal{C} = \mathbf{0}$.

Definition 2.10 (Tensor subspace [22], [39]): Given a set $\{\mathcal{D}_{(1)}, \ldots, \mathcal{D}_{(p)}\} \subseteq \mathbb{S}$ whose elements are linearly independent, the set

$$\mathbb{K}^{\mathbf{L}} = \{\mathcal{Y} | \mathcal{Y} = \mathcal{D} *_{\mathbf{L}} \mathcal{C}, \ \forall \mathcal{C} \in \mathbb{R}^{p \times 1 \times n_3}\} \tag{7}$$

is called a tensor subspace of dimension $\dim(\mathbb{K}^{\mathbf{L}}) = p$. The tensors $\mathcal{D}_{(1)}, \ldots, \mathcal{D}_{(p)}$ form the basis spanning $\mathbb{K}^{\mathbf{L}}$.

Based on the definition of a tensor subspace, we now clarify how data may be distributed with respect to such subspaces. In the simplest case, given a tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, all samples lie in the same tensor subspace: there exists a basis tensor $\mathcal{D} \in \mathbb{R}^{n_1 \times p \times n_3}$ such that every sample frontal slice can be written as $\mathcal{X}(:, i, :) = \mathcal{D}_{(i)} *_L \mathcal{C}_{(i)}$, $i = 1, \ldots, n_2$, where $\mathcal{C} \in \mathbb{R}^{p \times n_2 \times n_3}$ contains the coefficient tensors corresponding to each slice. This describes a *single subspace* model in which one shared low-dimensional structure explains the entire dataset.

However, in many practical scenarios, real multi-dimensional data are not well modeled by a single subspace. Instead, the data may be better described as lying in a union of multiple independent tensor subspaces, where different subsets of the data are generated from distinct subspaces with their own bases. Formally, the *multiple subspaces* can be written as $\mathcal{X} = [\mathcal{X}_1, \ldots, \mathcal{X}_k]$, $\mathcal{X}_j(:, i_j, :) \in \mathbb{K}_j^{\mathbf{L}}$, where each $\mathbb{K}_j^{\mathbf{L}}$ is a low-dimensional tensor subspace spanned by its own basis $\mathcal{D}_j \in \mathbb{R}^{n_1 \times p_j \times n_3}$. That is, for each subset $\mathcal{X}_j$, every slice satisfies $\mathcal{X}_j(:, i, :) = (\mathcal{D}_j)_{(i)} *_L (\mathcal{C}_j)_{(i)}$, $i = 1, \ldots, m_j$, with $\mathcal{C}_j \in \mathbb{R}^{p_j \times m_j \times n_3}$ containing the coefficients for the j-th subspace. It is worth noting that the assumption of independence among different subspaces is not strict. In fact, its vector counterpart has long been a standard assumption in sparse and low-rank data analysis [22].

III. Methodology

In this work, we propose a novel DEep Low-rank Tensor representAtion (DELTA), the outline of which is shown in Fig. 3, for multi-dimensional data recovery. This framework consists of two key components within the tensor algebraic skeleton of the t-SVD: (i) deep nonlinear transform (Section III-A) and (ii) the low-rank self-representation of multiple subspaces in the transform domain (Section III-B).

A. Multi-Head Attention-Based Group-Tube Deep Nonlinear Transform

The motivation behind designing a nonlinear transform in this work is to overcome the limitations of existing transforms in modeling complex multi-dimensional data. Linear transforms, such as DFT and discrete cosine transform (DCT) utilized in [8], [9], rely on fixed global bases and can only capture linear relationships between input and output, making them insufficient for representing the intricate nonlinear structures inherent in real-world data [10], [30], [31], [32]. Although nonlinear transforms, e.g., fully connected networks (FCNs) [10], [30], have been introduced to replace the linear transform for nonlinear feature extraction, their reliance on local connections limits their ability to model long-range dependencies. Meanwhile, most existing transform-based methods consider single tube transforms (along the third dimension), neglecting the homogeneity along the first and second dimensions.

To capture long-range dependencies and diverse patterns across different subspaces of the multi-dimensional data, within the transform-based t-SVD tensor skeleton, we propose and develop two structurally symmetric deep 3D transforms to replace the linear transform $\mathbf{L}$ and its inverse $\mathbf{L}^{-1}$, respectively. Given a tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, we denote the forward and backward transforms as $\mathcal{F}$ and $\mathcal{B}$, respectively. Both $\mathcal{F}$ and $\mathcal{B}$ consist of two components: a 1D spectral/temporal transform that exploits the internal correlation of the global features within each tube, and a 2D spatial local transform that explores the correlation between neighboring tubes.

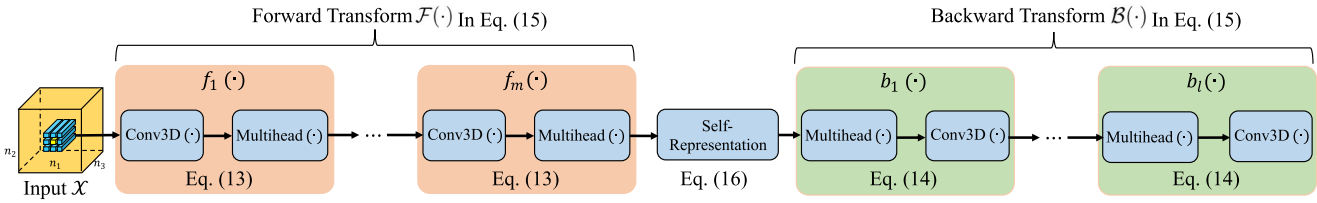

Fig. 3. Overall framework of DELTA. The input tensor $\mathcal{X}$ is first processed by the forward transform $\mathcal{F}(\cdot)$, where Conv3D$(\cdot)$ captures spatial homogeneity along the first and second modes, and the Multihead$(\cdot)$ models long-range dependencies along the third mode. The self-representation tensor enforces low-rankness to reveal multi-subspace structure. Finally, the backward transform $\mathcal{B}(\cdot)$ maps the low-rank representation back to the original data domain.

1) 1D Spectral/Temporal Transform: To extract long-range nonlinear dependencies, we adapt the multi-head attention mechanism [40] to implement the 1D spectral/temporal transform. Specifically, given a third-order tensor input $\mathcal{X}_{\text{in}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the *query*, *key*, and *value* are linearly projected from $\mathcal{X}_{\text{in}}$ as

$$\begin{cases} \mathbf{Q} \in \mathbb{R}^{n_3 \times n_1 n_2} = \mathbf{W}^q \texttt{unfold}_3(\mathcal{X}_{\text{in}}), \\ \mathbf{K} \in \mathbb{R}^{n_3 \times n_1 n_2} = \mathbf{W}^k \texttt{unfold}_3(\mathcal{X}_{\text{in}}), \\ \mathbf{V} \in \mathbb{R}^{n_3 \times n_1 n_2} = \mathbf{W}^v \texttt{unfold}_3(\mathcal{X}_{\text{in}}), \end{cases} \tag{8}$$

where $\mathbf{W}^q, \mathbf{W}^k, \mathbf{W}^v \in \mathbb{R}^{n_3 \times n_3}$ are the learnable weight matrices. Then, the self-attention is computed as

$$\texttt{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \tanh\left(\frac{\mathbf{Q}\mathbf{K}^\top}{n_1 n_2}\right)\mathbf{V}. \tag{9}$$

Here, we use $\tanh$ instead of the $\mathrm{softmax}$ function as the latter's sparse attention (caused by exponential amplification of high scores) may suppress critical high-frequency details, while $\tanh$ enables dense feature interactions through non-normalized outputs within $[-1, 1]$, avoiding over-sparsity and being helpful for identifying independent subspaces.

To further model the hierarchy and diverse patterns and benefit the subspace identification in the subsequent part, we further consider the multi-head attention and obtain *queries*, *keys*, and *values* as

$$\begin{cases} \mathbf{Q}_i \in \mathbb{R}^{\frac{n_3}{s} \times n_1 n_2} = \mathbf{W}_i^q \texttt{unfold}_3(\mathcal{X}_{\text{in}}), \\ \mathbf{K}_i \in \mathbb{R}^{\frac{n_3}{s} \times n_1 n_2} = \mathbf{W}_i^k \texttt{unfold}_3(\mathcal{X}_{\text{in}}), \\ \mathbf{V}_i \in \mathbb{R}^{\frac{n_3}{s} \times n_1 n_2} = \mathbf{W}_i^v \texttt{unfold}_3(\mathcal{X}_{\text{in}}), \end{cases} \tag{10}$$

where $\mathbf{W}_i^q, \mathbf{W}_i^k, \mathbf{W}_i^v \in \mathbb{R}^{\frac{n_3}{s} \times n_3}$ are learnable weight matrices, for $i = 1, 2, \ldots, s = \#\text{heads}$. Then, our 1D spectral/temporal transform $\text{MultiHead} : \mathbb{R}^{n_1 \times n_2 \times n_3} \to \mathbb{R}^{n_1 \times n_2 \times n_3}$ is constructed as

$$\texttt{MultiHead}(\mathcal{X}_{\text{in}}) = \texttt{fold}_3\left(\texttt{Concat}(\text{head}_1, \ldots, \text{head}_s)\right), \tag{11}$$

where each head is computed in parallel, i.e.,

$$\text{head}_i = \texttt{Attention}(\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i). \tag{12}$$

2) 2D Spatial Transform: While the 1D spectral/temporal transform with multi-head attention effectively captures long-range spectral or temporal dependencies and diverse patterns across different subspaces along the third mode, multi-dimensional data also exhibit local homogeneity, such as spatial smoothness in images. To complement this, we incorporate a 3D convolution [41] layer as a 2D spatial transform, which captures local spatial features and enhances the representation along the spatial dimensions. Specifically, given an input $\mathcal{X}_{\text{in}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the output of this 3D convolutional neural network (CNN) layer is given as

$$\text{Conv3D}(\mathcal{X}_{\text{in}}) \in \mathbb{R}^{n_1 \times n_2 \times n_3'} = \sigma\left(\mathcal{X}_{\text{in}} \circledast_{123} \mathbf{h}_k\right),$$

where $\circledast_{123}$ indicates the 3D convolution, $\mathbf{h}_k$s denote the convolution kernels of size $3 \times 3 \times 1$, σ is the nonlinear activation function, and n_3' is determined by the number of convolution kernels. In this work, we select the leaky rectified linear unit (Leaky ReLU) [45] as the nonlinear activation function.

3) Proposed Forward and Backward Transforms: Equipped with the above modules, we then formulate the forward and backward transforms. Denoting a single forward transform layer as

$$f_i(\cdot) = \text{MultiHead}\left(\text{Conv3D}(\cdot)\right), \tag{13}$$

and a single backward transform layer with a symmetrical structure as

$$b_i(\cdot) = \text{Conv3D}\left(\text{MultiHead}(\cdot)\right), \tag{14}$$

the group-tube deep nonlinear transform based forward transform $\mathcal{F} : \mathbb{R}^{n_1 \times n_2 \times n_3} \to \mathbb{R}^{n_1 \times n_2 \times \tilde{n}_3}$ and the backward transform $\mathcal{B} : \mathbb{R}^{n_1 \times n_2 \times \tilde{n}_3} \to \mathbb{R}^{n_1 \times n_2 \times n_3}$ are formulated as

$$\begin{cases} \mathcal{F}(\cdot) = f_m \circ f_{m-1} \circ \cdots \circ f_1(\cdot), \\ \mathcal{B}(\cdot) = b_l \circ b_{l-1} \circ \cdots \circ b_1(\cdot), \end{cases} \tag{15}$$

where m and l denote the number of forward and backward transform layers, respectively. From our ablation study in Section IV-D4, a small value of m and l is adequate in our framework. Generally, a larger $\tilde{n}_3$ in the transform domain could introduce redundancy and help to obtain a better low-rank representation [28], [30]. Therefore, we set $\tilde{n}_3 = 4n_3$ by adjusting the number of convolution kernels across this work.

This simple and novel design allows the proposed nonlinear transform to effectively model both long-range interactions along the third mode and local correlations between tubes, providing an accurate and compact representation of complex multi-dimensional data. To further validate its superior ability to represent the implicit low-rankness (i.e., the nonlinear inner correlations) of third-order tensor data, we show the low-rankness via the AccEgy in the transform domain with respect to different transforms in Fig. 4. From the AccEgy curves shown in Fig. 4, we can see that with the proposed forward and backward transforms,

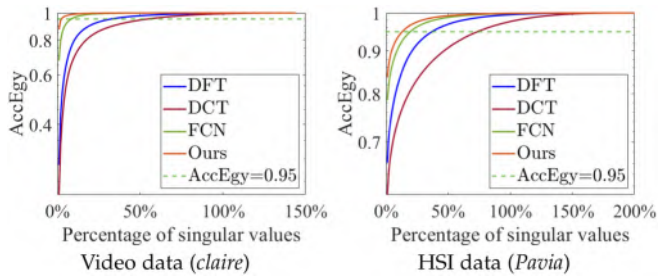

Fig. 4. The AccEgy of different types of tensor data with respect to different transforms within the t-SVD framework.

the AccEgy curve accumulates much more rapidly than others, showing that the energy concentrates on bigger singular values and thus yielding a more compact representation in the transform domain.

B. Deep Low-Rank Tensor Self-Representation

The key motivation is that most existing tensor low-rank recovery methods are limited to characterizing data within a single subspace distribution via regularizing the entire low-rankness of the tensor data. In practice, it is necessary to exploit the nature of multiple subspaces of real-world data. For example, the surveillance video illustrated in Fig. 2 shows that different moving objects in various temporal intervals result in the formation of multiple subspaces. Here we construct a low-rank tensor self-representation in the deep transform domain to tackle the multi-subspace structure. A more fine-grained characterization of multiple subspaces, rather than the overall low-dimensional space, in which the union of these subspaces resides, can generate better recovery results as the prior information of the data is depicted more meticulously, see our experimental results in Section IV.

1) Deep Tensor Self-Representation: In [36], [37], [46], self-representation methods have been studied under the matrix setting. The methods [37], [38] are to construct a matrix self-representation in the deep transform domain to handle situations where data points do not exactly reside in a union of *linear* subspaces. By using linear transforms under the t-SVD framework, tensor self-representation methods have also been developed, see [22], [39]. Here, the main contribution is to utilize nonlinear transforms developed in Section III-A to establish a deep tensor self-representation for the multi-subspace structure. With the deep nonlinear forward transform $\mathcal{F}$, the deep tensor self-representation is established in the deep transform domain as follows:

$$\mathcal{F}(\mathcal{X}) = \mathcal{F}(\mathcal{X}) \odot \hat{\mathcal{Z}}, \tag{16}$$

where $\hat{\mathcal{Z}} \in \mathbb{R}^{n_2 \times n_2 \times \tilde{n}_3}$ is the self-representation tensor in the transform domain. In the original domain, our deep tensor self-representation is formulated as $\mathcal{X} = \mathcal{B}(\mathcal{F}(\mathcal{X}) \odot \hat{\mathcal{Z}})$.

Remark 3.1: When the data tensor $\mathcal{X}$ is corrupted with outliers or noise, we need to consider an additional error tensor $\mathcal{E}$ in the original domain. We note that adding $\mathcal{E}$ in the original domain does not affect the self-representation format of (16) in the transform domain. Therefore, there is no need to minimize both distances in the transform domain and original domain as the deep transforms $\mathcal{F}$ and $\mathcal{B}$ are not ensured to be invertible. This is different from the matrix or linear t-prod based tensor self-representation cases.

Note that $\hat{\mathcal{Z}}$ is defined in the transform domain. According to Definition 2.7, the tubal-rank of the self-representation tensor in the original domain (potential exists) is equivalent to $\max_k \text{rank}(\hat{\mathcal{Z}}^{(k)})$. In the following, we analyze the frontal-slice-wise low-rank property of $\hat{\mathcal{Z}}$.

2) Low-Rankness of Representation Coefficients: We start with the linear-transform based t-SVD case and provide the following theorem.

Theorem 3.1: Given the tensor data $\mathcal{X}$, which is partitioned as $\mathcal{X} = [\mathcal{X}_1, \ldots, \mathcal{X}_k]$, where the lateral slices of each block $\mathcal{X}_j \in \mathbb{R}^{n_1 \times m_j \times n_3}$ are drawn from tensor subspace $\mathbb{K}_j^{\mathbf{L}}$ with $\dim(\mathbb{K}_j^{\mathbf{L}}) = p_j$. If the subspaces $\{\mathbb{K}_1^{\mathbf{L}}, \ldots, \mathbb{K}_k^{\mathbf{L}}\}$ are independent, the optimal minimizer $\mathcal{Z}^*$ of the following problem

$$\min_{\mathcal{Z}} \|\mathcal{Z}\|_{\text{L-TNN}} \quad \text{s.t. } \mathcal{X} = \mathcal{X} *_{\mathbf{L}} \mathcal{Z}$$

is block-diagonal and satisfies $\text{rank}_t(\mathcal{Z}^*) = \sum_{j=1}^k p_j$.

The proof of Theorem 3.1, provided in Supplementary Material, is an extension of the proof of Theorem 3 in [22], with figuring out the correlation between the rank of $\mathcal{Z}$ and the dimensionality of subspaces under linear invertible transform $\mathbf{L}$. By the specific design of the linear transform-based t-SVD, the above theorem is valid in both the original and transform domains (with the linear transform $\mathbf{L}$).

Then, we show that Theorem 3.1 can be generalized for deep nonlinear tensor self-representation. To accomplish this, the definition of the tensor subspace should be defined with respect to the deep nonlinear transforms $\mathcal{F}$ and $\mathcal{B}$.

For a linear invertible transform matrix $\mathbf{L} \in \mathbb{C}^{n_3 \times n_3}$, the tensor subspace definition in Definition 2.9 can be equivalently reformulated in the transform domain as

$$\mathbb{K}^{\mathbf{L}} = \{\mathcal{Y} \mid \mathcal{Y} \times_3 \mathbf{L} = (\mathcal{D} \times_3 \mathbf{L}) \odot \hat{\mathcal{C}},\ \forall \hat{\mathcal{C}} \in \mathbb{R}^{p \times 1 \times n_3}\}, \tag{17}$$

where $\hat{\mathcal{C}} = \mathcal{C} \times_3 \mathbf{L}$. An implicit requirement in (17) is that $\mathcal{Y} \times_3 \mathbf{L} \times_3 \mathbf{L}^{-1} = \mathcal{Y}$. To extend this formulation beyond linear transforms, we replace $\mathbf{L}$ and $\mathbf{L}^{-1}$ by a pair of nonlinear mappings $\mathcal{F}$ and $\mathcal{B}$, and assume (idealized) $\mathcal{B}(\mathcal{F}(\mathcal{X})) = \mathcal{X}$ holds. Then, we generalize the tensor subspace as

$$\mathbb{K}^{\mathcal{F},\mathcal{B}} = \{\mathcal{Y} = \mathcal{B}(\mathcal{F}(\mathcal{Y})) | \mathcal{F}(\mathcal{Y}) = \mathcal{F}(\mathcal{D}) \odot \hat{\mathcal{C}},\ \forall \hat{\mathcal{C}} \in \mathbb{R}^{p \times 1 \times \tilde{n}_3}\}, \tag{18}$$

where tensors $\mathcal{F}(\mathcal{D}_{(1)}), \ldots, \mathcal{F}(\mathcal{D}_{(p)})$ form the basis spanning $\mathbb{K}^{\mathcal{F},\mathcal{B}}$ in the transform domain. Ideally, one would require $\mathcal{B}(\mathcal{F}(\mathcal{X})) = \mathcal{X}$, but since $\mathcal{F}$ and $\mathcal{B}$ are not guaranteed to be invertible in practice, this identity should be regarded as a modeling generalization rather than a strict equality.

Remark 3.2: We assume that the deep nonlinear transforms $\mathcal{F}$ and $\mathcal{B}$ satisfy $\mathcal{B}\big(\mathcal{F}(\mathcal{X})\big) = \mathcal{X}$ theoretically. In practice, it is difficult to guarantee such semi-invertibility for arbitrary tensors. However, under our self-supervised learning scheme (see next subsection), it suffices to require approximate semi-invertibility within a local neighborhood of each target tensor.

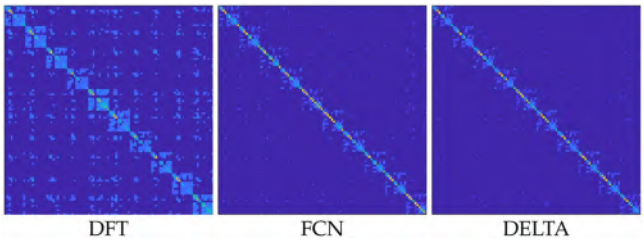

Fig. 5. The learned block-diagonal structures by TLRR, FCN, and DELTA on Face dataset (*Yale B*) of the size $48 \times 100 \times 42$. We show the sample similarity matrices $\tilde{\mathbf{Z}}$, which is obtained by $\tilde{\mathbf{Z}} = \frac{1}{2\tilde{n}_3}\sum_{i=1}^{\tilde{n}_3}(|\hat{\mathcal{Z}}(:,:,i)| + |\hat{\mathcal{Z}}(:,:,i)|^T)$.

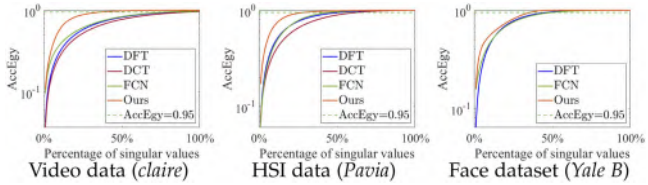

Fig. 6. The AccEgy of the representation coefficient tensor with respect to different transforms under the t-SVD framework.

Benefiting from the representation capacity of deep transforms, this requirement can be well achieved.

Note that since $\mathcal{F}$ and $\mathcal{B}$ are nonlinear, the subspaces defined in the transform domain would correspond to more complex manifolds in the original domain. For the purpose of theoretical formulation and algorithmic design, it is adequate to model them as linear subspaces in the transform domain, which provides a unified and tractable framework for analysis. We are now ready to generalize Theorem 3.1 to the deep nonlinear setting.

Theorem 3.2: Given the tensor data $\mathcal{X}$ and nonlinear transforms $\mathcal{F}$ and $\mathcal{B}$ such that $\mathcal{B}(\mathcal{F}(\mathcal{X})) = \mathcal{X}$, let $\hat{\mathcal{X}} = \mathcal{F}(\mathcal{X})$ be partitioned as $\hat{\mathcal{X}} = [\hat{\mathcal{X}}_1, \ldots, \hat{\mathcal{X}}_k]$, where the lateral slices of each block $\hat{\mathcal{X}}_j$ are drawn from a tensor subspace $\mathbb{K}_j^{\mathcal{F},\mathcal{B}} \subseteq \mathbb{K}^{\mathcal{F},\mathcal{B}}$ in the transform domain, with $\dim(\mathbb{K}_j^{\mathcal{F},\mathcal{B}}) = p_j$. If the subspaces $\{\mathbb{K}_1^{\mathcal{F},\mathcal{B}}, \ldots, \mathbb{K}_k^{\mathcal{F},\mathcal{B}}\}$ are independent, the optimal minimizer $\hat{\mathcal{Z}}^*$ of the following problem

$$\min_{\hat{\mathcal{Z}}} \sum_{k=1}^{\tilde{n}_3} \left\| \hat{\mathcal{Z}}^{(k)} \right\|_* \quad \text{s.t.} \quad \hat{\mathcal{X}} = \hat{\mathcal{X}} \odot \hat{\mathcal{Z}}$$

is block-diagonal and satisfies $\text{rank}_t(\hat{\mathcal{Z}}^*) = \sum_{j=1}^{k} p_j$.

The proof of Theorem 3.2 can be found in Supplementary Material.

Remark 3.3: Theorem 3.2 reveals one key property of the deep self-representation coefficient tensor $\hat{\mathcal{Z}}$: its frontal-slice-wise rank[2] equals the sum of the dimensions of the underlying subspaces in the transform domain. In practice, many real-world tensor datasets, such as videos, HSI data, and face datasets naturally occupy a union of low-dimensional subspaces. For instance, Fig. 5 illustrates the block-diagonal structure learned by our method on the *Yale B* face dataset, where images of the same person occupy a single low-rank block. Within each block, face images from one individual are highly correlated and thus low-rank. As in [22], the subspace-independence assumption is not strict; even if all samples lie in a single subspace, the frontal-slice-wise rank of $\hat{\mathcal{Z}}$ still matches the tubal-rank (dimension of the single subspace) of the data tensor. In Fig. 6, we illustrate the AccEgy of representation coefficients in the transform domain learned on a video, an HSI, and the Yale B face dataset. We can see that, across different data types and transforms, the representation coefficients clearly display a pronounced low-rank structure in the transform domain. Therefore, the low-rankness of $\hat{\mathcal{Z}}$ depends on the structure of $\mathcal{X}$, and boosting the low-rankness of $\hat{\mathcal{Z}}$ promotes the low-rankness of the original tensor $\mathcal{X}$ by restricting each subspace.

To achieve this, we minimize the frontal-slice-wise nuclear norm of $\hat{\mathcal{Z}}$ as

$$\sum_{k=1}^{\tilde{n}_3} \|(\hat{\mathcal{Z}})^{(k)}\|_*, \tag{19}$$

which can be seen as the tubal nuclear norm of the self-representation tensor in the original domain (potentially exists).

In real-world data, where samples lie on nonlinear, multi-subspace manifolds, this deep low-rank representation yields a block-diagonal coefficient tensor whose overall rank (the sum of the ranks of each diagonal block) is far smaller than treating all data as a single global subspace. When data occupy multiple independent subspaces, $\hat{\mathcal{Z}}$ naturally decomposes into disjoint diagonal blocks, one per subspace, with zero weights between blocks. It also ensures that each sample is reconstructed exclusively from its own subspace, eliminating redundant information and yielding more accurate recovery. This advantage stems from considering the low-rankness of the representation coefficient tensor rather than the data tensor. Moreover, as shown in Fig. 6, our learned nonlinear transform further concentrates singular-value energy in the coefficient tensor, tightening each subspace's representation and reinforcing its low-rank property. Consequently, this deep low-rank tensor representation more faithfully captures the nonlinear, multi-subspace manifolds present in real-world data.

C. DELTA for Multi-Dimensional Data Recovery

In this section, we describe how to construct the DELTA model for real-world multi-dimensional data recovery. The outline of DELTA is shown in Fig. 3. In summary, DELTA consists of two key components within the tensor algebraic skeleton of the t-SVD: (i) deep nonlinear transforms, which combine $\text{Conv3D}(\cdot)$ to capture spatial homogeneity along the first and second modes with $\text{Multihead}(\cdot)$ to capture long-range dependencies along the third mode, and (ii) the low-rank self-representation of multiple subspaces in the transform domain. Specifically, we first introduce the objective function and then present the optimization procedure.

1) Objective Function for Multi-Dimensional Data Recovery: Following (19) and equipping with the deep nonlinear transform and deep low-rank tensor self-representation, we

[2] The tubal-rank of a potential self-representation tensor $\mathcal{Z}$ corresponding to $\hat{\mathcal{Z}}$ (i.e. $\mathcal{F}(\mathcal{Z}) = \hat{\mathcal{Z}}$ or $\mathcal{B}(\hat{\mathcal{Z}}) = \mathcal{Z}$) in the transform domain.

construct the DEep Low-rank Tensor representAtion (DELTA) model for real-world multi-dimensional data recovery on the observed tensor $\mathcal{O} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, as

$$\min_{\mathcal{X},\hat{\mathcal{Z}},\Theta} \lambda \sum_{k=1}^{\tilde{n}_3} \left\| \hat{\mathcal{Z}}^{(k)} \right\|_* + \gamma \left\| \mathcal{F}(\mathcal{X}) - \mathcal{F}(\mathcal{X}) \odot \hat{\mathcal{Z}} \right\|_F^2 + \mathcal{L}\left(\mathcal{B}(\mathcal{F}(\mathcal{X}) \odot \hat{\mathcal{Z}}), \mathcal{O} \right), \tag{20}$$

where $\Theta = \{\mathbf{W}_i^q, \mathbf{W}_i^k, \mathbf{W}_i^v, \mathbf{h}_k\}$ denotes the learnable parameters of deep nonlinear transform, $\hat{\mathcal{Z}} \in \mathbb{R}^{n_2 \times n_2 \times \tilde{n}_3}$ is the self-representation tensor in the deep transform domain, and λ, γ are nonnegative trade-off parameters.

The second term in Eq. (20) enforces that the transform-domain self-representation remains close to the true transformed signal, where the low-rank regularization on $\mathcal{Z}$ captures structural redundancy while retaining essential data details. However, as emphasized in Remarks 3.1 and 3.2, the deep transforms $\mathcal{F}$ and $\mathcal{B}$ are generally not guaranteed to be perfectly invertible. To mitigate this, we introduce the fidelity term $\mathcal{L}(\mathcal{B}(\mathcal{F}(\mathcal{X})), \mathcal{O})$, which directly penalizes reconstruction errors in the original domain and is tailored to the degradation process in each specific application.

Remark 3.4: The fidelity term in DELTA serves two complementary purposes. First, it preserves the validity of the self-representation by aligning the learned transform-domain features with the original data, thereby reducing overfitting and distortions introduced by nonlinear mappings. Second, it compensates for the possible irreversibility of the deep transforms $\mathcal{F}$ and $\mathcal{B}$, working jointly with the low-rank constraint on $\mathcal{Z}$ to balance structural regularity in the transform domain with fidelity to observations in the original domain. Although this introduces an additional weighting parameter, it is typically insensitive and can be set to a default value across tasks, with fine-tuning seldom required. Consequently, the fidelity term is indispensable for ensuring both robustness and practical applicability of DELTA.

For tensor completion, the fidelity term is defined over the observed entries and measured by the Frobenius norm. Specifically, if Ω denotes the index set of observed entries, the fidelity term $\mathcal{L}(\mathcal{B}(\mathcal{F}(\mathcal{X}) \odot \hat{\mathcal{Z}}), \mathcal{O})$ is set as

$$\frac{1}{2} \left\| \mathcal{P}_\Omega \left(\mathcal{B}(\mathcal{F}(\mathcal{X}) \odot \hat{\mathcal{Z}}) \right) - \mathcal{O} \right\|_F^2, \tag{21}$$

where $\mathcal{P}_\Omega(\cdot)$ is the projection function that keeps the elements in the observed set Ω and makes others zero.

For robust tensor completion, the observed entries may not only be missing but also corrupted by large, sparse outliers. To handle this, we adopt an l_1 norm based fidelity term as

$$\left\| \mathcal{P}_\Omega \left(\mathcal{B}(\mathcal{F}(\mathcal{X}) \odot \hat{\mathcal{Z}}) \right) - \mathcal{O} \right\|_1. \tag{22}$$

For spectral snapshot imaging, given the observed $\mathbf{O} \in \mathbb{R}^{n_1 \times n_2}$. Then the fidelity term is defined as

$$\frac{1}{2} \left\| \sum_{k=1}^{n_3} \mathbf{M}^{(k)} \odot \left(\mathcal{B}(\widehat{\mathcal{F}(\mathcal{X})}) \right)^{(k)} - \mathbf{O} \right\|_F^2, \tag{23}$$

Algorithm 1: DELTA for Multi-Dimensional Data Recovery.

Input: The observed tensor $\mathcal{O}$; trade-off parameters λ and γ; maximum number of iterations $t_{\max}$.

1: **Initialization:** $\mathcal{X} = \text{Init}(\mathcal{O})$, $t = 0$.
2: **while** $t < t_{\max}$ **do**
3: $t = t + 1$;
4: Update $\{\Theta, \mathcal{X}, \hat{\mathcal{Z}}\}$ via minimizing (20) using Adam [47];
5: **end while**

Output: The recovered tensor $\mathcal{B}(\mathcal{F}(\mathcal{X}) \odot \hat{\mathcal{Z}})$.

where $\mathbf{M}^{(k)} \in \mathbb{R}^{n_1 \times n_2}$, constituted of 0 and 1, denotes the k-th coding mask and $\odot$ denotes the element-wise product.

2) Optimization Procedure: After initializing the latent tensor with the observation $\mathcal{O}$, we jointly optimize all learnable parameters in Θ, the self-representation tensor $\hat{\mathcal{Z}}$, and estimate data $\mathcal{X}$ via Adam [47]. The complete procedure is detailed in Algorithm 1. Because the objective in Eq. (20) is minimized directly on the degraded measurement, the entire process is fully self-supervised (i.e., a zero-shot learning manner), requiring no ground-truth reference during training.

Although the nonlinear transforms $\mathcal{F}$ and $\mathcal{B}$ are applied to $\mathcal{X}$, the model remains fully differentiable. Therefore, gradients with respect to $\mathcal{X}$ can be efficiently computed via backpropagation through the deep transforms. This allows $\mathcal{X}$, $\hat{\mathcal{Z}}$, and Θ to be jointly and iteratively updated in each optimization step. At every iteration, Adam updates the network parameters and self-representation tensor, and then refines $\mathcal{X}$ accordingly. The final restored tensor is obtained via the operation $\mathcal{B}(\mathcal{F}(\mathcal{X}) \odot \hat{\mathcal{Z}})$. This unified and fully differentiable formulation enables efficient self-supervised learning tailored to the characteristics of each specific degraded input.

Fig. 7 illustrates the model and network structure of our method, focusing on enhancing the low-rankness of the representation tensor rather than the entire data, allowing for the depiction of deep multi-subspaces.

Remark 3.5: Compared to matrix-based self-expression methods, tensor-based self-representation intrinsically preserves the multi-dimensional structure of data by operating within the tensor algebra framework. This allows the model to capture higher-order correlations and complex interdependencies across multiple modes, such as spatial, temporal, and spectral dimensions, simultaneously. For example, in multi-channel video or hyperspectral data, tensor self-representation can exploit spatial-temporal consistency and spectral correlations more effectively than treating data as independent matrices. Consequently, tensor formulations provide richer and more accurate representations, which lead to improved recovery and analysis performance, especially in scenarios where data exhibits inherent multi-subspace and multi-way relationships.

Moreover, the proposed DELTA model can be reduced to classic shallow representations under specific conditions:

i) When $\mathcal{F}$ and $\mathcal{B}$ are set as the discrete Fourier transform (DFT) matrix and its inverse DFT matrix, respectively,

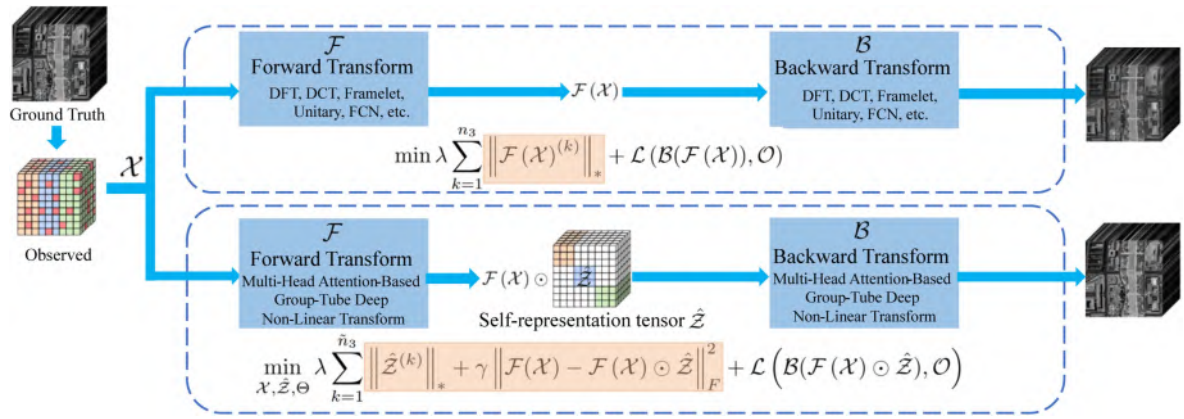

Fig. 7. The flowchart of our method. The upper row shows the previous methods, which directly minimize the nuclear norm in the transform domain. The bottom row shows our method that minimizes the nuclear norm of the self-representation tensor $\hat{\mathcal{Z}}$.

DELTA reduces to the tensor low-rank representation (TLRR) method [22], which explores the multi-subspace structure of the data using a linear transform.

ii) When $\hat{\mathcal{Z}}$ is set as the identity matrix, and $\mathcal{F}$ and $\mathcal{B}$ are set as fully connected neural networks, DELTA reduces into the S2NTNN method [30], which minimizes the nuclear norm of the frontal slices of the nonlinearly transformed tensor.

iii) When $\mathcal{F}$ and $\mathcal{B}$ are set as the DFT matrix and its inverse DFT matrix, $\hat{\mathcal{Z}}$ is set as the identity matrix, and minimizing the nuclear norm of the frontal slices of the data tensor, DELTA reduces into the traditional Tensor Nuclear Norm (TNN) method [8].

IV. Numerical Experiments

In this section, we validate the effectiveness and versatility of our proposed DELTA framework by applying it to three challenging multi-dimensional data recovery tasks: tensor completion (TC), robust tensor completion (RTC), and spectral snapshot imaging (SSI). The adaptability of DELTA is demonstrated by tailoring the fidelity term, $\mathcal{L}(\cdot,\cdot)$, in Eq. (20) to suit the specific noise model of each application.

Parameters and Network Configuration: In our method, the parameters λ and γ in (20) are empirically selected from the candidate sets $\{0.1, 1, 10, 100\}$ and $\{0.01, 0.1, 1, 10\}$, respectively, to ensure optimal performance across diverse data and tasks. For the deep nonlinear transforms, the learnable parameters $\Theta = \{\mathbf{W}_i^q, \mathbf{W}_i^k, \mathbf{W}_i^v, \mathbf{h}_k\}$ were initialized from a standard normal distribution. Unless otherwise specified, the layers of $\mathcal{F}$ and $\mathcal{B}$ for an input tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ were consistently set as $m = l = 1$ and an expanded channel dimension of $\tilde{n}_3 = 4n_3$. The optimization was performed for a maximum of $t_{\max} = 3100, 1600, 1600$ iterations (respectively for TC, RTC, and SSI) using a learning rate of 0.002. For the TC and RTC tasks, which involve inpainting missing entries, we initialized the tensor $\mathcal{X}$ using linear interpolation [53]. This strategy provides a reasonable and structured starting point that can accelerate convergence. For the SSI task, a simple back-projection of the compressed measurement [54] was used as the initial estimate.

All experiments were conducted on a platform equipped with an Intel (R) Core (TM) i5-10500 CPU, an NVIDIA RTX 3090 GPU, and 36 GB of RAM. The implementations of our method and other Python-based competitors were executed on PyTorch 1.10.0, leveraging both CPU and GPU capabilities. All MATLAB-based methods were run on MATLAB R2022b using only the CPU. To quantitatively assess the reconstruction quality, we employ a set of standard evaluation metrics. For all tasks, we report the peak signal-to-noise ratio (PSNR) and the structural similarity (SSIM) [55], where higher values indicate superior performance. For tasks involving multispectral (MSI) and hyperspectral (HSI) data, we additionally compute the spectral angle mapper (SAM) [56]. A lower SAM value signifies a smaller spectral distortion and thus a more accurate spectral reconstruction.

A. Tensor Completion

In this experiment, we evaluate the performance of our proposed method on the task of tensor completion (TC), where the goal is to recover a full tensor from a sparse set of observed entries. Our evaluation is conducted on four diverse types of data, namely, 31 multispectral images (MSIs) from the CAVE dataset,[3] 23 videos from the ASU Video Trace Library,[4] a 3D brain MRI scan from the BrainWeb database,[5] and a tensor constructed from 100 images of 10 individuals from the Yale B dataset.[6] We investigate two distinct missing data scenarios. First, on the MSI dataset, we simulate a **random missing** scenario, where individual pixels are randomly sampled with rates (SR) of $\{0.05, 0.1, 0.15, 0.2, 0.25\}$. Second, we consider a more challenging case of **tube-wise missing**, in which tubes along the third dimension ($1 \times 1 \times n_3$ fibers) are randomly removed. This scenario simulates structured data loss in real-world applications. For instance, in video surveillance, a damaged sensor may drop an entire pixel trajectory (resulting in a missing tube of pixel

[3] https://cave.cs.columbia.edu/repository/Multispectral
[4] http://trace.eas.asu.edu/yuv/index.html
[5] https://brainweb.bic.mni.mcgill.ca/brainweb/selection_normal.html
[6] http://cvc.cs.yale.edu/cvc/projects/yalefacesB/yalefacesB.html

TABLE I
QUANTITATIVE EVALUATION OF TENSOR COMPLETION METHODS ON THE CAVE MULTISPECTRAL IMAGE DATASET. THE TABLE PRESENTS AVERAGE PSNR, SSIM, AND SAM VALUES ON 31 IMAGES (EACH OF THE SIZE $512 \times 512 \times 31$) FOR BOTH RANDOM AND TUBE-WISE MISSING SCENARIOS. THE **BEST** AND SECOND-BEST RESULTS ARE HIGHLIGHTED IN **BOLD** AND UNDERLINED, RESPECTIVELY

Missing	Sampling Rate	0.05			0.1			0.15			0.2			0.25			Time
Scenario	Method	PSNR	SSIM	SAM	PSNR	SSIM	SAM	PSNR	SSIM	SAM	PSNR	SSIM	SAM	PSNR	SSIM	SAM	(s)
Random Missing	Observed	14.51	0.231	78.1	14.74	0.266	73.8	14.99	0.299	69.5	15.25	0.330	65.4	15.53	0.359	61.6	—
	HaLRTC [7]	26.15	0.795	14.4	30.05	0.865	9.4	32.51	0.902	7.4	34.33	0.925	6.3	35.82	0.941	5.5	32
	TNN [8]	30.24	0.805	14.5	36.68	0.922	7.7	39.57	0.954	5.8	41.44	0.968	4.8	43.02	0.977	4.1	212
	TLRR [22]	32.00	0.881	15.7	35.24	0.933	11.7	37.50	0.956	9.2	39.15	0.967	7.5	40.37	0.974	6.3	794
	HLRTF [10]	35.83	0.928	6.7	40.00	0.968	4.6	42.62	0.980	3.8	44.20	0.986	3.2	45.39	0.989	2.9	39
	S2NTNN [30]	33.54	0.899	12.0	37.19	0.940	7.9	39.76	0.958	6.0	41.89	0.969	4.8	43.65	0.978	3.8	312
	CoNoT [31]	37.20	0.947	6.8	41.05	0.973	4.5	43.42	0.982	3.4	45.29	0.987	2.9	46.77	0.991	2.5	279
	TWTC [48]	30.19	0.780	14.4	31.91	0.812	12.6	32.74	0.826	11.9	33.50	0.840	11.3	34.01	0.853	10.4	592
	TCTV [49]	34.55	0.934	5.7	38.77	0.966	4.3	41.40	0.978	3.6	43.37	0.985	3.1	44.98	0.988	2.7	367
	TNN-G [32]	33.67	0.849	16.3	38.00	0.917	10.1	39.94	0.942	7.5	40.98	0.954	6.2	41.68	0.960	5.6	789
	KBR [50]	37.25	0.934	5.9	42.60	0.974	3.9	45.04	0.984	3.2	46.79	0.989	2.8	48.17	0.992	2.5	822
	SALTS [51]	39.27	0.959	4.9	44.01	0.984	3.5	46.63	**0.991**	2.9	48.45	**0.994**	2.6	**49.85**	**0.995**	2.3	2946
	LRTFR [52]	35.63	0.902	7.7	37.69	0.929	6.7	39.11	0.944	6.0	39.73	0.949	5.8	40.72	0.957	5.3	129
	DELTA	**39.96**	**0.964**	**4.5**	**44.50**	**0.985**	**3.1**	**46.92**	**0.991**	**2.6**	**48.56**	0.993	**2.3**	49.72	**0.995**	**2.0**	249
Tube-wise Random Missing	Observed	14.51	0.231	85.4	14.74	0.266	81.0	14.99	0.299	76.5	15.25	0.330	72.0	15.53	0.359	67.5	—
	HaLRTC [7]	20.92	0.644	14.8	25.84	0.751	10.3	28.43	0.811	8.3	30.24	0.850	7.0	31.69	0.879	6.2	46
	TNN [8]	23.43	0.605	13.6	26.82	0.710	10.0	29.02	0.775	8.1	30.70	0.820	6.8	32.10	0.854	5.8	124
	TLRR [22]	26.89	0.752	12.9	29.47	0.844	10.2	31.20	0.888	8.4	32.51	0.915	7.2	33.59	0.932	6.3	991
	HLRTF [10]	29.28	0.835	9.5	31.25	0.878	7.1	32.66	0.901	6.0	33.77	0.918	5.2	34.65	0.930	4.6	39
	S2NTNN [30]	26.93	0.755	11.8	29.66	0.847	8.6	31.45	0.891	6.9	32.82	0.918	5.8	33.95	0.935	5.0	275
	CoNoT [31]	28.01	0.800	10.8	30.57	0.868	7.9	32.38	0.902	6.1	33.73	0.921	5.3	34.92	0.936	4.5	283
	TWTC [48]	24.70	0.564	16.5	26.61	0.660	13.5	28.15	0.724	11.1	29.36	0.768	9.5	30.52	0.805	8.2	527
	TCTV [49]	26.98	0.736	25.6	29.43	0.813	20.9	31.11	0.843	18.3	32.26	0.869	16.4	33.24	0.890	14.8	367
	TNN-G [32]	26.90	0.748	12.4	29.49	0.838	9.6	31.22	0.882	7.8	32.55	0.909	6.5	33.64	0.927	5.6	813
	KBR [50]	17.80	0.334	22.3	19.10	0.429	16.3	20.51	0.524	12.5	22.02	0.611	9.8	23.81	0.694	7.8	810
	SALTS [51]	22.55	0.655	10.8	27.90	0.786	7.2	31.14	0.855	5.7	33.52	0.896	4.7	35.11	0.921	4.0	2896
	LRTFR [52]	27.03	0.690	11.0	29.80	0.775	8.5	31.53	0.820	7.4	32.90	0.854	6.4	34.15	0.882	5.6	129
	DELTA	**30.79**	**0.869**	**6.8**	**33.40**	**0.896**	**5.1**	**35.75**	**0.927**	**4.2**	**37.42**	**0.944**	**3.6**	**38.89**	**0.955**	**3.2**	276

values across time). In hyperspectral imaging, a malfunctioning detector could yield no readings for a particular spatial location across all spectral bands (one pixel's spectral signature is entirely missing).

1) Compared Methods: We benchmark DELTA against a comprehensive suite of twelve representative tensor completion methods: a Tucker decomposition based method HaLRTC [7], a DFT transform-based TNN [8], a tensor low-rank representation method TLRR [22], two nonlinear transform based methods, HLRTF [10] and S2NTNN [30], a coupled nonlinear transform based method CoNoT [31], a tensor wheel decomposition based method TWTC [48], a fusing low-rankness and smoothness method TCTV [49], a group-tube transform based method TNN-G [32], a Kronecker-basis representation based method KBR [50], a continuous Tucker decomposition based method LRTFR [52] and a self-adaptive learnable transform based method SALTS [51].

2) Quantitative Analysis: The quantitative results, detailed in Tables I–IV, illustrate the state-of-the-art performance of our DELTA framework. On the CAVE MSI dataset under random missing conditions, DELTA is highly competitive, consistently ranking in the top two alongside SALTS [51] across all sampling rates while being over 11 times faster. More critically, its superiority becomes unequivocal in the more difficult tube-wise missing scenario. Here, DELTA consistently dominates across different types of data, though the primary competitor for the runner-up position varies. For instance, on MSIs at SR $= 0.25$, DELTA surpasses the second-best method, SALTS, by a substantial 3.78 dB in PSNR. This trend of clear dominance extends to the other datasets. On MRI at SR $= 0.1$, our DELTA outperforms

TABLE II
QUANTITATIVE COMPARISON FOR TENSOR COMPLETION ON MRI DATA ($143 \times 179 \times 121$) WITH TUBE-WISE MISSING ENTRIES. THE **BEST** AND SECOND-BEST RESULTS ARE HIGHLIGHTED IN **BOLD** AND UNDERLINED, RESPECTIVELY

Sampling Rate	0.1		0.2		0.3		Time
Method	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	(s)
Observed	10.58	0.316	11.10	0.344	11.68	0.374	—
HaLRTC [7]	16.17	0.261	19.96	0.460	23.05	0.619	5
TNN [8]	18.96	0.278	22.49	0.460	25.52	0.615	54
TLRR [22]	18.94	0.426	19.96	0.509	22.47	0.594	217
HLRTF [10]	17.88	0.378	24.87	0.688	26.43	0.763	23
S2NTNN [30]	18.92	0.449	20.00	0.530	21.53	0.591	55
CoNoT [31]	18.49	0.415	19.31	0.480	23.48	0.614	85
TWTC [48]	20.10	0.321	22.63	0.492	23.75	0.571	105
TCTV [49]	24.16	0.619	27.23	0.744	29.72	0.818	96
TNN-G [32]	18.95	0.425	19.96	0.507	20.74	0.553	244
KBR [50]	17.76	0.240	25.53	0.611	29.73	0.781	527
SALTS [51]	22.44	0.616	26.85	0.773	30.43	0.822	1440
LRTFR [52]	22.01	0.455	28.37	0.777	31.27	0.831	121
DELTA	**25.60**	**0.699**	**29.97**	**0.893**	**32.48**	**0.932**	234

the runner-up, TCTV [49], by a significant margin of 1.44 dB in PSNR, while on Yale B face images at SR $= 0.5$, it again leads the second-place method, LRTFR [52], by nearly 1.0 dB. Meanwhile, on video data at SR $= 0.3$, our DELTA leads the second-place method, SALTS, by nearly 1.0 dB. It is noteworthy that some methods that perform well under random missing conditions, such as KBR [50] and SALTS [51], see a significant performance degradation in the tube-wise scenario. This performance discrepancy can be attributed to that such structural missing affects the learning of key components in their model. For KBR [50], it distorts the learning of the specific factor matrix

TABLE III
QUANTITATIVE COMPARISON FOR TENSOR COMPLETION ON THE YALE B FACE DATASET ($48 \times 100 \times 42$) WITH TUBE-WISE MISSING ENTRIES. THE **BEST** AND SECOND-BEST RESULTS ARE HIGHLIGHTED IN **BOLD** AND UNDERLINED, RESPECTIVELY

Sampling Rate	0.1		0.3		0.5		Time
Method	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	(s)
Observed	7.52	0.035	8.51	0.096	10.01	0.184	—
HaLRTC [7]	9.88	0.118	18.20	0.538	23.17	0.790	1
TNN [8]	15.66	0.252	21.26	0.652	26.09	0.854	5
TLRR [22]	18.04	0.387	20.97	0.644	23.50	0.785	19
HLRTF [10]	19.59	0.510	23.27	0.746	26.12	0.852	9
S2NTNN [30]	17.91	0.409	20.92	0.654	23.46	0.786	22
CoNoT [31]	17.27	0.326	19.37	0.519	21.49	0.672	11
TWTC [48]	16.41	0.295	21.30	0.672	26.57	0.871	15
TCTV [49]	17.04	0.399	20.78	0.624	22.99	0.739	22
TNN-G [32]	18.04	0.398	20.96	0.651	23.46	0.784	16
KBR [50]	15.98	0.248	21.92	0.686	25.57	0.853	22
SALTS [51]	17.18	0.354	23.50	0.756	26.85	0.864	101
LRTFR [52]	19.08	0.494	24.58	0.810	28.01	0.907	78
DELTA	**20.21**	**0.574**	**24.97**	**0.844**	**28.97**	**0.929**	59

TABLE IV
QUANTITATIVE COMPARISON OF TENSOR COMPLETION METHODS ON VIDEO DATA WITH TUBE-WISE MISSING ENTRIES. THE VALUES ARE THE AVERAGE VALUES OVER 23 VIDEOS, COMPRISING 17 AT $144 \times 176 \times 100$ AND 6 AT $288 \times 352 \times 50$ RESOLUTION. THE **BEST** AND SECOND-BEST RESULTS ARE HIGHLIGHTED IN **BOLD** AND UNDERLINED, RESPECTIVELY

Sampling Rate	0.1		0.2		0.3		Time
Method	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	(s)
Observed	6.50	0.019	7.01	0.034	7.58	0.051	—
HaLRTC [7]	17.47	0.357	20.90	0.551	23.12	0.675	12
TNN [8]	20.14	0.436	22.29	0.569	24.20	0.679	67
TLRR [22]	22.29	0.610	24.10	0.727	25.43	0.790	317
HLRTF [10]	22.30	0.598	24.19	0.725	25.66	0.794	20
S2NTNN [30]	22.17	0.576	23.88	0.711	25.18	0.770	80
CoNoT [31]	22.40	0.613	24.14	0.684	25.52	0.751	91
TWTC [48]	18.98	0.333	20.46	0.455	21.96	0.568	226
TCTV [49]	21.36	0.590	23.01	0.659	24.16	0.709	190
TNN-G [32]	22.60	0.630	24.33	0.731	25.96	0.796	171
KBR [50]	17.51	0.321	20.54	0.497	23.94	0.669	249
SALTS [51]	19.64	0.450	23.58	0.682	26.24	0.802	1238
LRTFR [52]	19.83	0.374	22.90	0.577	25.33	0.711	79
DELTA	**22.78**	**0.636**	**25.23**	**0.761**	**27.21**	**0.834**	182

and core tensor in Tucker decomposition, while for SALTS [51], it hampers the learning of the transform in the specific mode. In contrast, our DELTA, equipped with a learnable group-tube transform and multi-subspace representation, is more robust under structured missing scenarios.

3) Qualitative Analysis: The visual comparisons for the tube-wise missing scenario, presented in Fig. 8, provide compelling qualitative evidence of DELTA's superior performance. Across all four types of data, most competing methods produce reconstructions suffering from significant artifacts, such as heavy blurring, loss of fine texture, or residual structural noise. For example, in the MSI and video reconstructions, many methods fail to preserve sharp edges and intricate details, resulting in overly smooth or distorted images. Similarly, when applied to MRI data, competing techniques often sacrifice fine anatomical structures for noise removal. In stark contrast, our DELTA method consistently delivers reconstructions with exceptional visual fidelity. It effectively restores sharp details, accurate colors, and complex textures, from the anatomical structures in the brain MRI to the facial features and clothing patterns in the Yale B and video datasets. In all test cases, the output from DELTA is visually closest to the ground truth, highlighting its robustness in preserving critical image features.

B. Snapshot Spectral Imaging

We conduct our experiments for SSI using two hyperspectral images (HSIs) (*WDC mall*[7] and *Pavia*[8]), as well as three multispectral images (MSIs), *flowers*, *toy*, and *watercolors*. To mitigate the computational burden associated with the DeSCI method [57] and to accommodate the unique input requirements of STFormer [54], all datasets are truncated, retaining only 8 spectral bands. The sampling ratios (SRs) of the masks are set to 0.1, 0.3, and 0.5.

1) Compared Methods: We compare DELTA for SSI with the following methods: two generalized alternating projection (GAP)-based methods, GAP-TV [58] and DeSCI [57]; a plug-and-play method incorporating a deep neural network, PnP [59]; the combined TV and PnP method, PnP-TV [60]; the combined 3DTV and PnP method, PnP-3DTV; a nonlinear transform-based method, HLRTF [10]; a continuous Tucker decomposition based method, LRTFR [52]. For a broader perspective, we also include STFormer [54], a fully supervised state-of-the-art deep learning method. It is important to note that STFormer relies on extensive training with paired ground-truth data, whereas DELTA and the other benchmarks operate in an unsupervised, model-based fashion. Therefore, STFormer is included here primarily as a high-performance reference for the qualitative visual evaluation.

2) Results and Analysis: The results for the SSI task, presented in Table V and Fig. 9, unequivocally establish the state-of-the-art performance of our DELTA framework. Quantitatively, DELTA dominates across all sampling rates on both HSI and MSI datasets. The performance gap is often substantial. For instance, on HSI data, DELTA achieves a remarkable PSNR gain of nearly 5 dB against the second-best method, LRTFR [52], at SR $= 0.3$. This numerical superiority is visually corroborated in Fig. 9. While most competing model-based methods produce reconstructions with noticeable blurring and loss of fine detail, DELTA restores sharp edges and intricate textures with exceptional fidelity. Most notably, the performance of our unsupervised DELTA framework is visually on par with that of the fully supervised method STFormer [54]. This result is particularly significant, as it demonstrates that our method not only outperforms other model-based techniques by a large margin but also closes the gap with supervised approaches, offering a powerful and more flexible state-of-the-art solution for SSI.

C. Robust Tensor Completion

This part evaluates our DELTA on the demanding task of robust tensor completion (RTC), where the objective is to recover a tensor from incomplete observations that are further corrupted by sparse noise. We use two MSIs (*beads* and *flowers*) and three

[7] https://engineering.purdue.edu/~biehl/MultiSpec/hyperspectral.html
[8] http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes

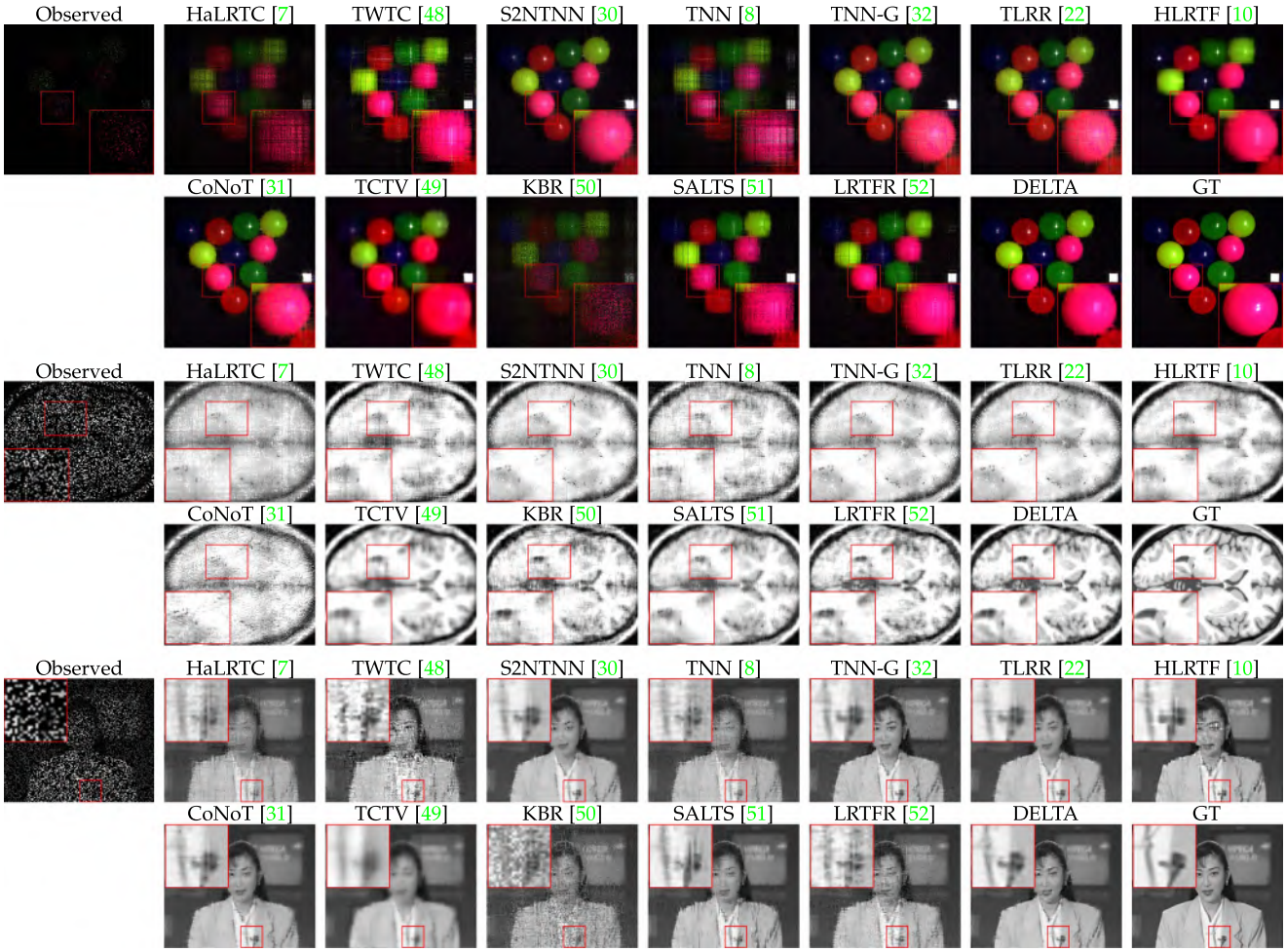

Fig. 8. Visual comparison of tensor completion results under tube-wise missing conditions across different data modalities. **Top**: Pseudo-color reconstructions of MSI (*superballs*, SR=0.05). **Second**: Frontal slices from an MRI volume (*Brain*, SR=0.2). **Bottom**: Reconstructed 11th frame from a video sequence (*akiyo*, SR=0.3).

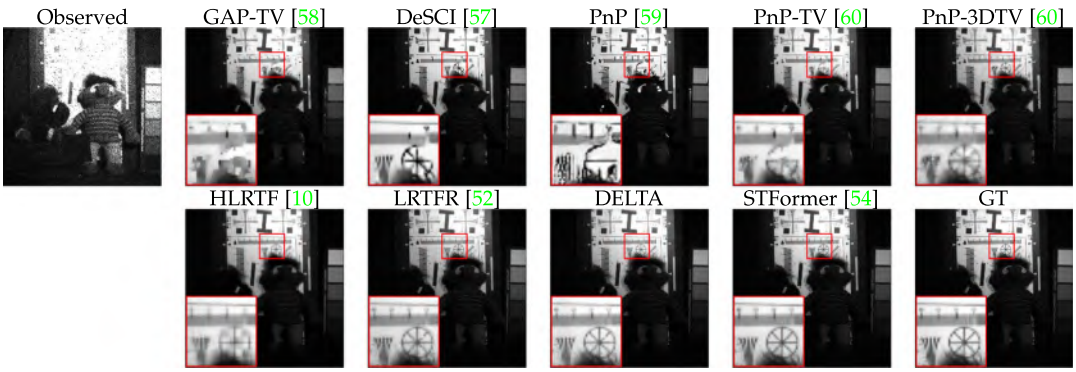

Fig. 9. Visual comparison of SSI reconstruction methods on the MSI *toy* dataset with SR $= 0.5$.

videos (*akiyo*, *carphone*, and *salesman*). We first subsample these data with SRs $= 0.1$, 0.2, and 0.3 to obtain incomplete tensors, and then add sparse salt-and-pepper noise to the observed entries with a fixed noise ratio of 0.1.

1) Compared Methods: We benchmark DELTA against six methods designed for RTC: ℓ_1-SNN [61], an ℓ_1-regularized sum of nuclear norm method, ℓ_1-TRNN [62], an ℓ_1-regularized tensor ring nuclear norm method, C2FRTRC [63] and its variant HQTRC, which are M-estimator-based robust tensor ring recovery methods, the capped Frobenius norm-based CFN-RTC [64], and the continuous Tucker decomposition based method LRTFR [52].

TABLE V
AVERAGE QUANTITATIVE RESULTS FOR SPECTRAL SNAPSHOT IMAGING (SSI). THE TOP SET OF ROWS PRESENTS THE PERFORMANCE ON HSIS (*WDC MALL* AND *PAVIA*, BOTH $200 \times 200 \times 8$), WHILE THE BOTTOM SET SHOWS RESULTS ON MSIS (*FLOWERS*, *TOY*, AND *WATERCOLORS*, ALL $256 \times 256 \times 8$). THE **BEST** AND SECOND-BEST RESULTS ARE HIGHLIGHTED IN **BOLD** AND UNDERLINED, RESPECTIVELY

Sampling Rate	0.1			0.3			0.5			Time
Method	PSNR	SSIM	SAM	PSNR	SSIM	SAM	PSNR	SSIM	SAM	(s)
GAP-TV [58]	23.75	0.581	9.9	27.36	0.815	8.4	31.21	0.928	6.7	1211
DeSCI [57]	23.21	0.559	8.9	26.92	0.827	7.9	28.10	0.889	6.7	6185
PnP [59]	23.14	0.613	15.7	23.54	0.685	15.8	22.08	0.632	18.9	1
PnP-TV [60]	24.37	0.616	9.1	26.97	0.798	8.8	28.45	0.871	9.0	9
PnP-3DTV [60]	27.39	0.784	5.6	30.48	0.902	6.3	31.80	0.934	6.6	4
HLRTF [10]	28.40	0.814	**2.1**	31.14	0.881	**1.9**	34.03	0.937	1.7	7
LRTFR [52]	29.20	0.857	**2.1**	33.81	0.928	**1.9**	39.87	0.967	1.7	80
DELTA	**29.54**	**0.881**	**2.1**	**38.59**	**0.984**	**1.9**	**43.79**	**0.995**	**1.6**	36
GAP-TV [58]	22.11	0.723	9.1	27.54	0.885	5.6	29.43	0.925	5.9	831
DeSCI [57]	25.20	0.823	5.8	28.06	0.904	4.7	28.72	0.918	4.8	8487
PnP [59]	20.64	0.791	8.4	25.54	0.861	6.6	24.62	0.839	7.8	1
PnP-TV [60]	25.31	0.816	5.6	28.75	0.906	4.5	30.10	0.931	4.5	9
PnP-3DTV [60]	27.04	0.843	13.1	29.54	0.900	11.1	30.11	0.914	11.8	4
HLRTF [10]	27.33	0.847	6.5	30.19	0.913	5.7	31.77	0.932	5.6	7
LRTFR [52]	28.67	0.882	5.4	31.70	0.935	5.2	32.13	0.940	4.9	85
DELTA	**29.52**	**0.891**	**4.6**	**32.29**	**0.944**	**4.4**	**33.82**	**0.951**	**4.2**	48

TABLE VI
THE **AVERAGE** QUANTITATIVE RESULTS ON VIDEO *akiyo*, *carphone*, AND *salesman* (OF THE SIZE $144 \times 176 \times 100$) FOR **RTC**. THE **BEST** AND SECOND-BEST RESULTS ARE HIGHLIGHTED IN **BOLD** AND UNDERLINED, RESPECTIVELY

Sampling Rate	0.1		0.2		0.3		Time
Method	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	(s)
Observed	6.82	0.013	7.26	0.023	7.74	0.035	—
ℓ_1-SNN [61]	21.92	0.669	26.03	0.823	28.92	0.896	61
ℓ_1-TRNN [62]	24.70	0.719	24.76	0.723	26.20	0.779	362
C2FRTRC [63]	25.99	0.820	30.21	0.916	32.00	0.934	525
HQTRC [63]	24.11	0.720	26.93	0.822	28.99	0.869	62
CFNRTC [64]	26.81	0.796	27.62	0.825	27.52	0.832	916
LRTFR [52]	26.84	0.827	26.95	0.835	28.05	0.843	86
DELTA	**29.60**	**0.913**	**31.41**	**0.936**	**34.55**	**0.962**	144

TABLE VII
THE **AVERAGE** QUANTITATIVE RESULTS ON MSIS *balloons* AND *beads* (OF THE SIZE $256 \times 256 \times 31$) FOR **RTC**. THE **BEST** AND SECOND-BEST RESULTS ARE HIGHLIGHTED IN **BOLD** AND UNDERLINED, RESPECTIVELY

Sampling Rate	0.1			0.2			0.3			Time
Method	PSNR	SSIM	SAM	PSNR	SSIM	SAM	PSNR	SSIM	SAM	(s)
Observed	11.07	0.031	75.1	11.30	0.046	71.1	11.55	0.061	67.5	—
ℓ_1-SNN [61]	21.28	0.468	41.7	28.93	0.790	12.3	31.73	0.873	9.4	38
ℓ_1-TRNN [62]	27.25	0.672	18.8	27.50	0.683	18.6	29.17	0.733	16.3	381
C2FRTRC [63]	28.80	0.786	11.2	33.33	0.914	7.2	36.98	0.962	4.9	446
HQTRC [63]	26.32	0.696	14.4	29.69	0.800	11.2	32.23	0.863	9.4	42
CFNRTC [64]	26.31	0.752	13.5	26.46	0.765	12.6	26.61	0.767	12.5	813
LRTFR [52]	30.35	0.856	9.4	34.21	0.936	6.6	36.58	0.966	5.2	76
DELTA	**33.33**	**0.925**	**6.4**	**36.68**	**0.964**	**4.8**	**39.08**	**0.977**	**3.9**	65

2) Results and Analysis: As illustrated by the quantitative and qualitative results in Tables VI-VII and Fig. 10, DELTA achieves state-of-the-art performance in the challenging RTC setting. The numerical results in the tables show that DELTA consistently outperforms all competing methods across both data types and all sampling rates. The performance margin is often significant; on video data at SR $= 0.3$, DELTA achieves a PSNR of 34.55 dB, surpassing the runner-up C2FRTRC [63] by 2.55 dB while being over 4 times faster. A similar trend is observed on MSI data, where DELTA leads the second-best method by 2.1 dB. This robust performance stems from our model's ability to accurately characterize the underlying structure, thus effectively disentangling the low-rank tensor structure from the sparse noise. The visual results in Fig. 10 corroborate this. While many competing methods either fail to remove all noise artifacts or excessively blur the image (e.g., C2FRTRC [64] and LRTFR [52]), DELTA successfully suppresses the heavy noise while simultaneously restoring fine structural details and sharp edges. This supports the framework's robustness and applicability to real-world scenarios with complex corruptions.

TABLE VIII
RESULTS OF DIFFERENT SELF-REPRESENTATION (SR) FORMATS ON THE *AKIYO* VIDEO ($144 \times 176 \times 100$, SR=0.1). THE **BEST** AND SECOND-BEST RESULTS ARE HIGHLIGHTED IN **BOLD** AND UNDERLINED, RESPECTIVELY

SR format	Tensor	Nonlinear	PSNR	SSIM
$\mathcal{X} = \mathcal{X} \odot \mathcal{Z}$	—	—	21.44	0.447
$\mathbf{X}_{(3)} = \mathbf{X}_{(3)}\mathbf{Z}$	—	—	23.16	0.515
$\mathcal{X} = \mathcal{X} *_{\text{DFT}} \mathcal{Z}$ [22]	✓	—	29.20	0.910
$\mathbf{X}_\theta = \mathbf{X}_\theta\mathbf{Z}$ [37]	—	✓	28.83	0.916
$\mathcal{F}(\mathcal{X}) = \mathcal{F}(\mathcal{X}) \odot \hat{\mathcal{Z}}$	✓	✓	**35.25**	**0.975**

D. Ablation Study and Discussions

In this part, we conduct a series of ablation studies and further analyses to deconstruct our DELTA framework. We investigate the contributions of its core components, analyze the impact of key architectural choices and hyperparameters, and evaluate its scalability and computational complexity.

1) Analysis of Self-Representation Format: First, we validate the effectiveness of the deep tensor self-representation format in (16), i.e., $\mathcal{F}(\mathcal{X}) = \mathcal{F}(\mathcal{X}) \odot \hat{\mathcal{Z}}$. We take the video data *akiyo* of the size $144 \times 176 \times 100$ with a random sampling rate of 0.1 for testing. Four alternatives are considered. The first one omits the nonlinear transform in (16) and directly applies slice-wise matrix self-representation, i.e., $\mathcal{X} = \mathcal{X} \odot \mathcal{Z}$. The second one directly unfolds the data tensor $\mathcal{X}$ along the third mode and adopts the matrix self-representation as $\mathbf{X}_{(3)} = \mathbf{X}_{(3)}\mathbf{Z}$. The third one is TLRR [22], which uses the DFT-based tensor self-representation, i.e., $\mathcal{X} = \mathcal{X} *_{\text{DFT}} \mathcal{Z}$. The first three variants are linear. Then, we consider a nonlinear variant proposed in [37], which projects the data matrix $\mathbf{X}_{(3)}$ via an encoder as $\mathbf{X}_\theta$, where θ indicates the parameter of the encoder.[9] Then, the matrix self-representation in the nonlinear latent space is expressed as $\mathbf{X}_\theta = \mathbf{X}_\theta\mathbf{Z}$.

Table VIII reports the quantitative results. The results clearly indicate that both nonlinearity and a tensor-native framework are

[9] We adopt the same convolutional structure of the encoder and decoder in [37]. The encoding step is accomplished by feeding frontal slices of the video data into the convolutional neural network (CNN) and reshaping the results into a matrix.

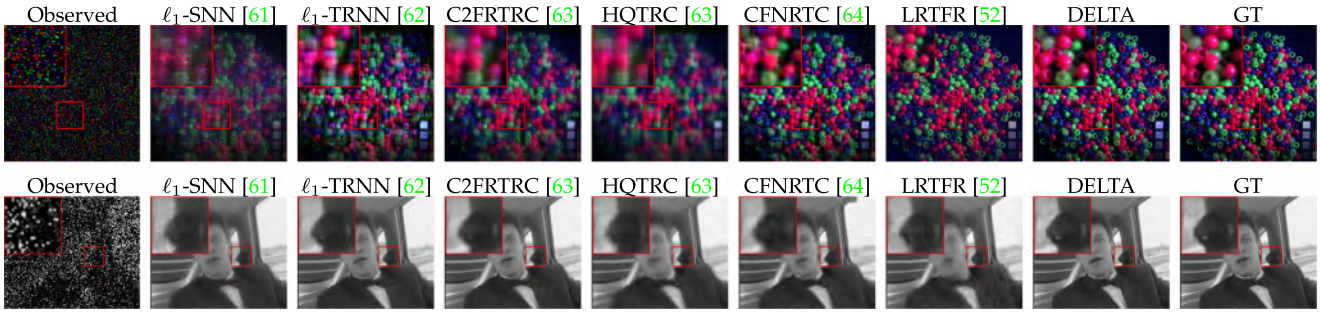

Fig. 10. Restoration results of RTC by different methods respectively on MSI *beads* (composed of the 23-th, 15-th, 1-st bands) with SR $= 0.1$ and the 50-th frame of video *carphone* with SR $= 0.3$.

TABLE IX
EFFECTIVENESS OF ADDING THE SELF-REPRESENTATION MODULE TO VARIOUS TRANSFORMS. TESTED ON MSI *BALLOONS* ($512 \times 512 \times 31$, SR=0.1) FOR RANDOM MISSING **TC**

Transforms	w/wo Self-Representation	PSNR	SSIM	Time (s)
DFT	wo (TNN [8])	38.87	0.967	205
	w	39.78	0.969	826
DCT	wo (TNN-DCT [9])	39.54	0.968	195
	w	43.42	0.975	231
FCN	wo (S2NTNN [30])	39.94	0.954	308
	w	46.01	0.989	321
CNN	wo (CoNoT [31])	44.09	0.986	274
	w	46.23	0.992	175
$\mathcal{F}$ & $\mathcal{B}$	wo	<u>46.80</u>	<u>0.992</u>	240
	w	**49.17**	**0.995**	246

TABLE X
ANALYSIS OF THE SELF-REPRESENTATION TENSOR $\mathcal{Z}$. TESTED ON MSI *BALLOONS* (OF THE SIZE $512 \times 512 \times 31$, SR=0.1) AND VIDEO *HIGHWAY* (OF THE SIZE $144 \times 176 \times 100$, SR=0.1) FOR RANDOM MISSING **TC**.. TOP: IMPACT OF INITIALIZATION. BOTTOM: A FACTORIZED TYPE-II ALTERNATIVE

Method		PSNR	SSIM	Time (s)
Initialization of $\hat{\mathcal{Z}}$ in DELTA	$i = 1$	45.14	0.990	247
	$i = 0$	**49.17**	**0.995**	245
	$i = 10^{-1}$	47.58	0.993	246
	$i = 10^{-2}$	48.86	0.994	248
	$i = 10^{-3}$	<u>48.96</u>	<u>0.995</u>	244
	$i \sim \exp(\mathcal{N}(0,1))$	48.93	0.994	246
Different types of $\hat{\mathcal{Z}}$	DELTA	<u>32.82</u>	<u>0.916</u>	92
	Type II	**33.26**	**0.921**	60

crucial. Naive matrix-based self-representation ($\mathbf{X}_{(3)} = \mathbf{X}_{(3)}\mathbf{Z}$) or slice-wise operations ($\mathcal{X} = \mathcal{X} \odot \mathcal{Z}$) yield poor performance. While existing tensor self-representation [22] or nonlinear matrix-based self-representation [37] methods offer significant improvements, our proposed formulation, which uniquely combines a tensor-centric framework with deep nonlinear transforms, achieves substantially better results, yielding a PSNR gain of over 6 dB against the next-best alternative.

2) Effect of the Self-Representation Module: Second, we evaluate the effect of the self-representation module under five transforms: two fixed (DFT [8], DCT [26]), two data-driven (FCN [30], CoNoT [31]), and our proposed nonlinear transforms $\mathcal{F}, \mathcal{B}$. Table IX reports results with and without SR. Under fixed operators, self-representation-based variants consistently outperform their TNN baselines, directly confirming the benefit of low-rank self-representation. Similar gains appear with data-driven operators. Moreover, with learnable nonlinear transforms $\mathcal{F}, \mathcal{B}$, DELTA achieves the best overall performance, highlighting both the validity of self-representation and the superiority of our nonlinear transform design.

3) Analysis of the Self-Representation Tensor $\hat{\mathcal{Z}}$: We further analyze the properties of the self-representation tensor $\hat{\mathcal{Z}}$. The top rows of Table X show the model's sensitivity to the initialization value, i, of $\hat{\mathcal{Z}}$. The performance is stable and near-optimal for initializations in the range $0 \leq i < 1$, with a significant performance drop only when $i \geq 1$. This suggests that a large initial value can obscure the original data features, whereas a small value allows the model to effectively learn the intrinsic structure.

To explore efficiency improvements, we also designed a factorized "Type II" self-representation, $\hat{\mathcal{Z}} = \hat{\mathcal{Z}}_1 \odot \hat{\mathcal{Z}}_2$, where $\hat{\mathcal{Z}}_1 \in \mathbb{R}^{n_2 \times r \times \tilde{n}_3}$ and $\hat{\mathcal{Z}}_2 \in \mathbb{R}^{r \times n_2 \times \tilde{n}_3}$ are smaller tensors. As shown in the bottom rows of Table X, this Type-II model can achieve slightly better performance with a faster runtime. However, we observed that its performance can be unstable and requires manual tuning of the inner rank r, making it a promising but less robust direction for future work. Our default DELTA model remains the more stable and reliable configuration.

4) Impact of Network Hyperparameters in Transforms: In this part, we analyze the impact of the key architectural choices within our nonlinear transforms, focusing on how different values of the head, layer, and kernel size, affect recovery accuracy and computational efficiency. In the default settings, we fixed certain parameters as follows: the number of layers (i.e., $m = l$ in (15)) is set to 1, the number of heads to 2, and the convolution kernel size to (1,3,3). For each of these configurations, we maintain the other hyperparameters consistent with those used in the default implementation. To deeply investigate the layers and the number of heads within our nonlinear transforms, we ablate without convolution and vary i) the number of heads in the multi-head attention and ii) the number of layers in the 1D transform. Subsequently, we examine the convolution operation by varying the kernel size.

Table XI reports the results. We can observe that increasing the number of layers yields better SSIM values, while the

TABLE XI
THE PARAMETERS ABLATION STUDY RESULTS FOR **TC WITH TUBE-WISE MISSING ENTRIES** ON VIDEO *AKIYO* WITH SR=0.1. THE **BEST** AND THE SECOND-BEST VALUES ARE HIGHLIGHTED

# Head	PSNR	SSIM	Time (s)	# Layer	PSNR	SSIM	Time (s)	Kernel Size	PSNR	SSIM	Time (s)
1	23.88	0.701	71	1	**24.34**	0.730	73	$1 \times 1 \times 1$	21.37	0.478	67
2	**24.34**	**0.730**	73	2	23.48	0.736	96	$3 \times 3 \times 1$	24.34	0.730	73
4	21.33	0.555	74	3	24.14	**0.767**	120	$5 \times 5 \times 1$	**24.45**	**0.735**	76
5	24.27	0.724	77	4	23.72	0.723	142	$3 \times 3 \times 3$	23.91	0.701	75
10	23.96	0.700	79	5	23.67	0.757	165	$3 \times 3 \times 5$	23.90	0.712	106
20	24.00	0.706	91	10	19.05	0.504	262	$5 \times 5 \times 5$	22.07	0.525	170

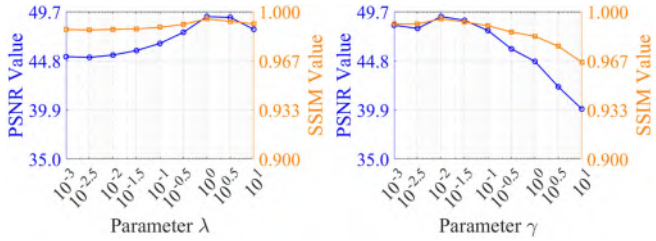

Fig. 11. PSNR and SSIM values with respect to different λ and γ for TC on the MSI *balloons* data (SR=0.1).

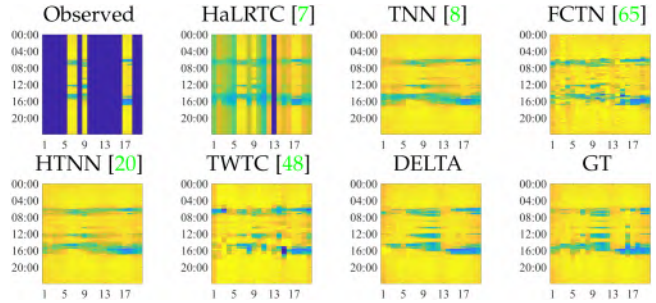

Fig. 12. Restoration result of higher-order completion by different methods respectively on traffic data with SR = 0.4 on the 7th day.

performance degrades when the transforms are too deep. This is because, when the training data is limited to the observed data, a deeper network is more prone to the vanishing gradient problem. This issue slows down the training process by causing the gradients of the loss function for the weights to become very small. Additionally, while increasing the size of the convolution kernel focuses on more localized information, it may lead to information disturbance when the kernel size is too large, resulting in decreased performance. Moreover, a larger convolution kernel size requires more computations.

5) Sensitivity to Regularization Parameters: We analyze the sensitivity of DELTA to the main regularization parameters, λ and γ, in (20). To test the effects of different values of them, we conduct experiments on MSI data *balloons* with the random sampling rate of 0.1. When testing one parameter, the other is fixed to its default value. We illustrate the PSNR and SSIM values with respect to different values of those parameters in Fig. 11. The analysis reveals that our model is robust and not overly sensitive to the precise setting of these hyperparameters. For the parameter λ, which balances the fidelity and regularization terms, the model achieves consistently high performance for values in $[10^{-1}, 10^{1}]$, with performance peaking near $\lambda = 1$. Similarly, for the self-representation parameter γ, the model is stable across a wide range of small values, maintaining near-optimal performance for any $\gamma \in [10^{-3}, 10^{-1}]$. Performance gracefully degrades only when γ becomes larger. The existence of these wide, stable windows for both parameters confirms the model's robustness and simplifies the tuning process, reducing the need for exhaustive hyperparameter searches.

TABLE XII
RESULTS FOR HIGHER-ORDER COMPLETION ON THE TRAFFIC DATA OF THE SIZE $60 \times 24 \times 20 \times 30$. THE **best** AND SECOND-BEST VALUES ARE HIGHLIGHTED

Sampling Rate	0.1		0.2		0.3		Time
Method	RMSE	MAPE	RMSE	MAPR	RMSE	MAPE	(s)
Observed	0.780	72.66	0.719	63.57	0.656	52.14	-
HaLRTC [7]	0.225	3155.9	0.071	6.0	0.065	4.86	4
TNN [8]	0.094	64.43	0.080	24.46	0.076	7.57	13
FCTN [65]	0.086	10.26	0.088	31.42	0.079	9.23	67
HTNN [20]	0.064	4.06	0.062	3.41	0.053	2.64	9
TWTC [48]	0.061	3.95	0.058	3.35	0.050	2.62	62
DELTA	**0.055**	**3.72**	**0.045**	**2.95**	**0.039**	**2.23**	153

6) Scalability to Higher-Order Tensors: Next, we test the scalability of our method to higher-order tensors and another type of multi-dimensional data. When dealing with 4th-order tensors, we employ the same strategy as HTNN [20], in which the transform along the third and fourth modes are decoupled. We decouple the transforms along the third and fourth modes of the tensor, adding a nonlinear transform along the fourth dimension. This approach allows us to scale without significantly increasing the number of parameters or imposing additional computational burdens. A fourth-order traffic data *Jan2019*,[10] which is provided by Grenoble Traffic Lab (GTL), is selected. The compared methods are listed as follows: a fully-connected tensor network based method FCTN-TC [65], a higher order t-SVD based method HTNN-FFT [20], HaLRTC [7], and TWTC [48]. We use root mean square error (RMSE) for the quantitative evaluation. Lower RMSE values refer to better performance. Fig. 12 illustrates the reconstruction results. Table XII shows the proposed method achieves the lowest RMSE values.

7) Choice of Normalization Function in Attention: We then investigate the effect of the normalization function used in (9). Specifically, we replace the standard $\mathrm{softmax}$ with $\tanh$ to examine its influence. The experiment is conducted on the MSI

[10] https://gtl.inrialpes.fr/data/gtl_data_jan2019.csv

TABLE XIII
STUDY OF THE NORMALIZATION FUNCTION IN (9) ON MSI *BALLOONS* (OF THE SIZE $512 \times 512 \times 31$, SR$= 0.1$). THE **BEST** AND THE SECOND-BEST VALUES ARE HIGHLIGHTED

Method	PSNR	SSIM	Time (s)
Softmax	47.07	0.991	245
tanh	**49.17**	**0.995**	245

TABLE XIV
THE COMPUTATIONAL COMPLEXITY PER EPOCH (ITERATION)

Method	3rd-order tensor	4th-order tensor
TNN (or HTNN) [20]	$\mathcal{O}(n^3(n+\log(n)))$	$\mathcal{O}(n^4(n+\log(n)))$
S2NTNN [30]	$\mathcal{O}(k^2n^4)$	$\mathcal{O}(k^2n^5)$
DELTA	$\mathcal{O}(c^2kn^4)$	$\mathcal{O}(c^2kn^5)$

balloons dataset with a sampling rate (SR) of 0.1 for the tensor completion task. The corresponding results are reported in Table XIII. As shown, the use of tanh leads to a clear improvement over Softmax in recovery performance, which can be attributed to the fact that tanh helps preserving critical high-frequency details without over-sparsity.

8) Computational Costs: For simplicity, we only consider one layer of the transform. Given a 3rd-order tensor $\mathcal{X} \in \mathbb{R}^{n\times n\times n}$ (or a 4th-order of the size $n \times n \times n \times n$), denoting k as the multiple of the length on specific dimension of the output enlarged by the 3D-CNN in DELTA, h as the head number in the multi-head attention layer, and $(1, c, c)$ as the kernel size of 3D-CNN, the computational complexities of TNN based methods are summarized in Table XIV.

V. CONCLUSION

We introduce a deep low-rank tensor representation for the recovery of real-world multi-dimensional data. First, the nonlinearity inherent in real-world tensor data is effectively captured using newly designed deep transforms. Then, within the transform-based t-SVD framework, we construct the self-representation tensor and minimize its nuclear norm, in the transform domain, to exploit the multi-subspace nature of real-world data. Experiments across various types of multi-dimensional tensor data and recovery tasks show that our method outperforms state-of-the-art approaches. Moreover, our method can be readily extended for higher-order tensors.

REFERENCES

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

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

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

[4] Y. Zhang and Y. Hu, "Dynamic cardiac MRI reconstruction using combined tensor nuclear norm and casorati matrix nuclear norm regularizations," in *Proc. Int. Symp. Biomed. Imag.*, 2022, pp. 1–4.

[5] T.-X. Jiang, M. K. Ng, J. Pan, and G.-J. Song, "Nonnegative low rank tensor approximations with multidimensional image applications," *Numerische Mathematik*, vol. 153, no. 1, pp. 141–170, 2023.

[6] 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.

[7] 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.

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

[9] C. Lu, X. Peng, and Y. Wei, "Low-rank tensor completion with a new tensor nuclear norm induced by invertible linear transforms," in *IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5996–6004.

[10] Y. Luo, X.-L. Zhao, D. Meng, and T.-X. Jiang, "HLRTF: Hierarchical low-rank tensor factorization for inverse problems in multi-dimensional imaging," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 19303–19312.

[11] J. D. Carroll and J.-J. Chang, "Analysis of individual differences in multidimensional scaling via an N-way generalization of 'eckart-young' decomposition," *Psychometrika*, vol. 35, no. 3, pp. 283–319, 1970.

[12] J. Douglas Carroll, S. Pruzansky, and J. B. Kruskal, "CANDELINC: A general approach to multidimensional analysis of many-way arrays with linear constraints on parameters," *Psychometrika*, vol. 45, pp. 3–24, 1980.

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

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

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

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

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

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

[19] 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.

[20] W. Qin, H. Wang, F. Zhang, J. Wang, X. Luo, and T. Huang, "Low-rank high-ordertensor completion with applications in visual data," *IEEE Trans. Image Process.*, vol. 31, pp. 2433–2448, 2022.

[21] 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.

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

[23] 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.

[24] 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.

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

[26] W.-H. Xu, X.-L. Zhao, and M. Ng, "A fast algorithm for cosine transform based tensor singular value decomposition," 2019, *arXiv:1902.03070*.

[27] 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.

[28] 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.

[29] T. Wu and J. Fan, "Smooth tensor product for tensor completion," *IEEE Trans. Image Process.*, vol. 33, pp. 6483–6496, 2024.

[30] Y.-S. Luo, X.-L. Zhao, T.-X. Jiang, Y. Chang, M. K. Ng, and C. Li, "Self-supervised nonlinear transform-based tensor nuclear norm for multi-dimensional image recovery," *IEEE Trans. Image Process.*, vol. 31, pp. 3793–3808, 2022.

[31] 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.

[32] B.-Z. Li, X.-L. Zhao, X. Zhang, T.-Y. Ji, X. Chen, and M. K. Ng, "A learnable group-tube transform induced tensor nuclear norm and its application for tensor completion," *SIAM J. Imag. Sci.*, vol. 16, no. 3, pp. 1370–1397, 2023.
[33] P. N. Belhumeur and D. J. Kriegman, "What is the set of images of an object under all possible illumination conditions?," *Int. J. Comput. Vis.*, vol. 28, no. 3, pp. 245–260, 1998.
[34] J. Ho, M.-H. Yang, J. Lim, K.-C. Lee, and D. Kriegman, "Clustering appearances of objects under varying illumination conditions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2003, vol. 1, pp. I–I.
[35] R. Vidal, Y. Ma, and S. Sastry, "Generalized principal component analysis (GPCA)," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 12, pp. 1945–1959, Dec. 2005.
[36] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171–184, Jan. 2013.
[37] P. Ji, T. Zhang, H. Li, M. Salzmann, and I. Reid, "Deep subspace clustering networks," in *Proc. Adv. Neural Inf. Process. Syst.*, pp. 23–32, 2017, vol. 30.
[38] P. Zhou, Y. Hou, and J. Feng, "Deep adversarial subspace clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1596–1604.
[39] 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.
[40] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, vol. 30, pp. 6000–6010.
[41] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4489–4497.
[42] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.
[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/CVF Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 3842–3849.
[44] 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.
[45] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. 30th Int. Conf. Mach. Learn., JMLR Workshop Conf. Proc.*, 2013, pp. 1–8.
[46] S. Wei and Z. Lin, "Analysis and improvement of low rank representation for subspace segmentation," 2011, *arXiv:1107.1561*.
[47] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic gradient descent," in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 1–15.
[48] Z.-C. Wu, T.-Z. Huang, L.-J. Deng, H.-X. Dou, and D. Meng, "Tensor wheel decomposition and its tensor completion application," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, vol. 35, pp. 27008–27020.
[49] H. Wang, J. Peng, W. Qin, J. Wang, and D. Meng, "Guaranteed tensor recovery fused low-rankness and smoothness," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 9, pp. 10990–11007, Sep. 2023.
[50] Q. Xie, Q. Zhao, D. Meng, and Z. Xu, "Kronecker-basis-representation based tensor sparsity and its applications to tensor recovery," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 8, pp. 1888–1902, Aug. 2018.
[51] T. Wu, B. Gao, J. Fan, J. Xue, and W. L. Woo, "Low-rank tensor completion based on self-adaptive learnable transforms," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 7, pp. 8826–8838, Jul. 2024.
[52] Y. Luo, X. Zhao, Z. Li, M. K. Ng, and D. Meng, "Low-rank tensor function representation for multi-dimensional data recovery," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 05, pp. 3351–3369, May 2024.
[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] L. Wang, M. Cao, Y. Zhong, and X. Yuan, "Spatial-temporal transformer for video snapshot compressive imaging," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 7, pp. 9072–9089, Jul. 2023.
[55] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[56] R. H. Yuhas, A. F. Goetz, and J. W. Boardman, "Discrimination among semi-arid landscape endmembers using the spectral angle mapper (SAM) algorithm," in *Proc. Jet Propulsion Lab. Airborne Geosci. Workshop*, pp. 147–149, 1992.
[57] Y. Liu, X. Yuan, J. Suo, D. J. Brady, and Q. Dai, "Rank minimization for snapshot compressive imaging," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 12, pp. 2990–3006, Dec. 2019.
[58] X. Yuan, "Generalized alternating projection based total variation minimization for compressive sensing," in *Proc. IEEE Int. Conf. Image Process.*, 2016, pp. 2539–2543.
[59] X. Yuan, Y. Liu, J. Suo, and Q. Dai, "Plug-and-play algorithms for large-scale snapshot compressive imaging," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 1447–1457.
[60] H. Qiu, Y. Wang, and D. Meng, "Effective snapshot compressive-spectral imaging via deep denoising and total variation priors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 9127–9136.
[61] D. Goldfarb and Z. Qin, "Robust low-rank tensor recovery: Models and algorithms," *SIAM J. Matrix Anal. Appl.*, vol. 35, no. 1, pp. 225–253, 2014.
[62] H. Huang, Y. Liu, Z. Long, and C. Zhu, "Robust low-rank tensor ring completion," *IEEE Trans. Comput. Imag.*, vol. 6, pp. 1117–1126, 2020.
[63] Y. He and G. K. Atia, "Coarse to fine two-stage approach to robust tensor completion of visual data," *IEEE Trans. Cybern.*, vol. 54, no. 1, pp. 136–149, Jan. 2024.
[64] X. P. Li, Z.-Y. Wang, Z.-L. Shi, H. C. So, and N. D. Sidiropoulos, "Robust tensor completion via capped Frobenius norm," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 7, pp. 9700–9712, Jul. 2024.
[65] 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, pp. 11071–11078.

Guo-Wei Yang is currently working toward the PhD degree with the School of Computing and Artificial Intelligence, Southwestern University of Finance and Economics. His research interests include tensor modeling and image processing.

Liqiao Yang received the MS degree in mathematics from the University of Electronic Science and Technology of China, Chengdu, in 2020, and the PhD degree in mathematics from the University of Macau, China, in 2024. She is currently a postdoctoral researcher with the Southwestern University of Finance and Economics. Her research focuses on quaternion modeling and algorithms for high-order data recovery.

Tai-Xiang Jiang (Member, IEEE) received the BS degree in mathematics and the PhD degree in applied mathematics from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2013 and 2019, respectively. He is currently a professor with the School of Computing and Artificial Intelligence, Southwestern University of Finance and Economics. His research interests include sparse and low-rank modeling, tensor decomposition, and multi-dimensional image processing.

Guisong Liu (Member, IEEE) received the BS degree in mechanics from Xi'an Jiao Tong University, Xi'an, China, in 1995, and the MS degree in automatics and the PhD degree in computer science from the University of Electronic Science and Technology of China, Chengdu, China, in 2000 and 2007, respectively. He was a visiting scholar with Humbolt University, Berlin, Germany, in 2015. He was a professor with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, in 2021. He is currently a professor and dean of the School of Computing and Artificial Intelligence, Southwestern University of Finance and Economics, Chengdu. He has filed more than 20 patents and authored or coauthored more than 70 scientific conference and journal papers. His research interests include pattern recognition, neural networks, and machine learning.

Michael K. Ng received the BSc and MPhil degrees from the University of Hong Kong, in 1990 and 1992, respectively, and the PhD degree from the Chinese University of Hong Kong, in 1995. He was a research fellow with Computer Sciences Laboratory, Australian National University, from 1995 to 1997, and assistant/associate professor with the University of Hong Kong, from 1997 to 2005. He was a professor/chair professor with the Department of Mathematics, Hong Kong Baptist University, from 2006 to 2019. He was the chair professor with the Research Division of Mathematical and Statistical Science, The University of Hong Kong, from 2019 to 2023. He is currently the chair professor of mathematics and chair professor in data science with Hong Kong Baptist University. His research interests include bioinformatics, image processing, scientific computing, and data mining. He is selected for the 2017 Class of Fellows of the Society for Industrial and Applied Mathematics. He was the recipient of Feng Kang Prize for his significant contributions in scientific computing. He is on the Editorial Board members of several international journals