Learnable Spatial-Spectral Transform-Based Tensor Nuclear Norm for Multi-Dimensional Visual Data Recovery

Sheng Liu, Jinsong Leng, Xi-Le Zhao, Haijin Zeng, *Graduate Student Member, IEEE*, Yao Wang, and Jing-Hua Yang

***Abstract*— Recently, transform-based tensor nuclear norm (TNN) methods have received increasing attention as a powerful tool for multi-dimensional visual data (color images, videos, and multispectral images, etc.) recovery. Especially, the redundant transform-based TNN achieves satisfactory recovery results, where the redundant transform along spectral mode can remarkably enhance the low-rankness of tensors. However, it suffers from expensive computational cost induced by the redundant transform. In this paper, we propose a learnable spatial-spectral transform-based TNN model for multi-dimensional visual data recovery, which not only enjoys better low-rankness capability but also allows us to design fast algorithms accompanying it. More specifically, we first project the large-scale original tensor to the small-scale intrinsic tensor via the learnable semi-orthogonal transforms along the spatial modes. Here, the semi-orthogonal transforms, serving as the key building block, can boost the spatial low-rankness and lead to a small-scale problem, which paves the way for designing fast algorithms. Secondly, to further boost the low-rankness, we apply the learnable redundant transform along the spectral mode to the small-scale intrinsic tensor. To tackle the proposed model, we apply an efficient proximal alternating minimization-based algorithm, which enjoys a theoretical convergence guarantee. Extensive experimental results on real-world data (color images, videos, and multispectral images) demonstrate that the proposed method outperforms state-of-the-art competitors in terms of evaluation metrics and running time.**

***Index Terms*— Tensor completion, semi-orthogonal transform, redundant transform, tensor nuclear norm, proximal alternating minimization algorithm.**

Manuscript received 24 March 2023; revised 27 June 2023; accepted 10 September 2023. Date of publication 18 September 2023; date of current version 9 May 2024. This work was supported in part by the NSFC under Grant 12371456, Grant 12171072, and Grant 62131005; in part by the National Key Research and Development Program of China under Grant 2020YFA0714001; and in part by the Open Research Fund Program of Data Recovery Key Laboratory of Sichuan Province under Grant DRN2302. This article was recommended by Associate Editor Y. Chen. *(Corresponding authors: Jinsong Leng; Xi-Le Zhao.)*

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

Haijin Zeng is with the Image Processing and Interpretation imec Research Group, Ghent University, 9000 Ghent, Belgium (e-mail: zeng_navy@163.com).

Yao Wang is with the Center for Intelligent Decision-Making and Machine Learning, School of Management, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: yao.s.wang@gmail.com).

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

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TCSVT.2023.3316279.

Digital Object Identifier 10.1109/TCSVT.2023.3316279

I. Introduction

TENSOR is a high-dimensional extension of vector/matrix, which provides a flexible and efficient mathematical representation for multi-dimensional visual data. Consequently, tensor representation plays an important role in real-world applications, e.g., image/video reconstruction and classification [1], [2], [3], [4], [5]. However, due to the limitations of the imaging device and environment, the collected multi-dimensional visual data is usually corrupted, e.g., video pixel missing [6], image graffiti [7], and thick cloud occlusion for remote sensing images [8], which hinders their subsequent applications [9], [10]. Recovering underlying data from the missing or corrupted partial observation, i.e., tensor completion, is a classical inverse problem in multi-dimensional image processing. Due to the non-uniqueness and instability of the solution to the inverse problem, the key to the recovery of multi-dimensional visual data is to efficiently characterize and leverage the prior knowledge.

Low-rankness is suggested to characterize the intrinsic structure of multi-dimensional visual data as the prior knowledge. The multi-dimensional visual data is generally self-correlated with low-rank structures, such as color images, videos, and multispectral images (MSIs). Thus, the low-rank tensor completion (LRTC) methods that exploit the underlying low-rank structure of data show promise for solving multi-dimensional visual data recovery problems [11], [12], [13], [14]. The LRTC method can be mathematically formulated as

$$\min_{\mathcal{X}} \operatorname{rank}_*(\mathcal{X}), \ \ s.t. \ \mathcal{X}_\Omega = \mathcal{O}_\Omega, \tag{1}$$

where $\mathcal{X}$ is the underlying tensor, $\mathcal{O}$ is the observed tensor, Ω is the index set of observed entries in $\mathcal{O}$, and $\operatorname{rank}_*(\cdot)$ is denoted as a specific type of tensor rank based on the corresponding rank assumption. The mainstream tensor ranks include CANDECOMP/PAEAFAC (CP) rank [15], Tucker rank [16], [17], tensor train rank [12], [18], tensor ring rank [19], [20], fully-connected tensor network rank [21], [22], and tensor tubal rank induced by tensor singular value decomposition (t-SVD) [23], [24].

The recent t-SVD has received increasing attention due to its excellent ability in capturing the widespread spatial-shifting correlations in real-world data [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33]. Kilmer et al. defined tensor tubal rank [23] and multi-rank [24] by using circulant algebra

with the framework of t-SVD. However, it is NP-hard to minimize the multi-rank of target tensors directly. To overcome this issue, Zhang et al. [25] employed tensor nuclear norm (TNN) as the convex surrogate of the tensor multi-rank and proposed a corresponding LRTC method. Lu et al. [26] defined an average rank and proposed the corresponding TNN, and further proved that this TNN is the convex envelope of the tensor average rank with the unit ball of the tensor spectral norm. Additionally, Wu et al. [28] proposed the tensor Qatar Riyal (QR) decomposition based on t-SVD, and employed the tensor QR to define TNN on small-scale tensors. In the t-SVD framework, the keystone is the orthogonal and predefined discrete Fourier transform (DFT) [23], which restricts the characterization of low-rankness of multi-dimensional visual data.

Many researchers have considered different transforms in t-SVD to characterize the low-rankness of tensors [34], [35], [36], [37], [38], [39]. Lu et al. [34] and Xu et al. [35] utilized the discrete cosine transform (DCT) and achieved the superior performance than the DFT. Kernfeld et al. [36] declared that a more generalized tensor-tensor product can be constructed by replacing the DFT with any linear transforms. Furthermore, Jiang et al. [37] suggested a semi-invertible framelet transform, which breaks the limitation of linear invertible transforms. However, the above transform matrices are data-independent and thus lack flexibility. To extend the flexibility of the transform matrices, it is necessary to design learning-based methods to make the transform matrices data-dependent [38], [39], [40], [41], [42]. Song et al. [38] suggested a unitary matrix that is generated by singular value decomposition (SVD) of a mode-3 unfolding matrix of the tensor. Jiang et al. [39] built a learnable redundant dictionary with low-rank coding as the transform matrix to define TNN named the dictionary-based TNN (DTNN) and achieved remarkable success in characterizing low-rankness. The success of DTNN lies in the redundant transform which is used to explore the low-rank structure of tensors. However, the redundant transform suffers from high computation cost, especially for large-scale tensors, where series SVDs are the main bottleneck.

To address the challenge computational issue, many fast algorithms were developed [43], [44], [45], [46], [47]. Zhang et al. [43] proposed a random t-SVD to solve this issue. They constructed a semi-orthogonal projection using a random tensor to project a large-scale tensor into a small-scale tensor, so that the SVD can be calculated on the small-scale tensor to reduce computational cost. Zhang et al. [44] designed a fast algorithm by bilateral random projections, which are utilized to constrain the tubal rank of the intrinsic tensor. Wang et al. [45] employed the random projection technique and the power of the block Krylov iteration to accurate frequent directions algorithm. By random projection, these fast algorithms have gained computational advantages at the cost of accuracy.

A natural question is can we reduce computational cost without sacrificing recovery performance under the t-SVD framework? To answer this question, we propose the learnable spatial-spectral transform-based tensor nuclear norm (LS2T2NN) model for LRTC. In the LS2T2NN model, we project the large-scale original tensor to the small-scale intrinsic tensor with learnable semi-orthogonal transforms along the spatial modes, which boosts the spatial low-rankness and allows us to design efficient optimization algorithms. There are many efficient algorithms (e.g., proximal alternating minimization (PAM)) available for solving nonconvex and nonsmooth optimization problems. To solve the proposed model, we apply PAM to design a solving algorithm with a theoretical convergence guarantee. The aforementioned idea is illustrated in Fig. 1.

TABLE I
NOTATIONS AND THE CORRESPONDING EXPLANATIONS

Notations	Explanations
x, $\boldsymbol{x}$, $\boldsymbol{X}$, $\mathcal{X}$	Scalar, vector, matrix, tensor.
x_{ijk}	The (i,j,k)-th entry of tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$.
$\mathrm{Tr}(\boldsymbol{X})$	The trace of $\boldsymbol{X} \in \mathbb{R}^{n \times n}$.
$\|\mathcal{X}\|_F$	The Frobenius norm of $\mathcal{X}$.
$\mathcal{X}(:,:,k), \boldsymbol{X}_k$	The k-th frontal slice of tensor $\mathcal{X}$.
$\mathcal{X}(i,j,:)$	The (i,j)-th tube of $\mathcal{X}$.
$\|\boldsymbol{X}\|_*$	The nuclear norm of $\boldsymbol{X}$.
$\mathrm{vec}(\boldsymbol{X})$	Stacking the columns of $\boldsymbol{X}$ into a vector.
$\mathrm{vec}^{-1}(\boldsymbol{X})$	The inverse operation of $\mathrm{vec}(\boldsymbol{X})$.
$\mathrm{unfold}_p(\mathcal{X}), \boldsymbol{X}_{(p)}$	The mode-p matricization of $\mathcal{X}$.
$\mathrm{fold}_p(\boldsymbol{X}_{(p)})$	The inverse operation of $\mathrm{unfold}_p(\mathcal{X})$.
$\boldsymbol{X}^\top$	The transpose of $\boldsymbol{X}$.
$\mathcal{X}^H$	The conjugate transpose of $\mathcal{X}$.

Our contributions are three-fold:

- Armed with the learnable semi-orthogonal transforms along the spatial modes and the redundant transform along the spectral mode, we propose the LS2T2NN model for multi-dimensional visual data recovery, which enjoys better low-rankness characterization and allows us to design efficient algorithms.
- To tackle the proposed model, we apply PAM to design a solving algorithm and establish the convergence guarantee.
- Extensive numerical experiments on videos, MSIs, and color images demonstrate that the proposed LS2T2NN has the superior performance compared with other state-of-the-art LRTC methods in terms of evaluation metrics and running time.

The rest of the paper is organized as follows. In Section II, we introduce some notations and preliminaries. In Section III, we propose the LS2T2NN model for LRTC and discuss the relationship between LS2T2NN and DTNN with theoretical analysis. In addition, we present an efficient PAM-based algorithm for LS2T2NN, which enjoys a theoretical convergence guarantee. The experimental results are illustrated and discussed in Section IV. The conclusion is drawn in Section V.

II. NOTATIONS AND PRELIMINARIES

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

Definition 1 (Mode-p product [48]): The mode-p product of a tensor $\mathcal{X}$ and a matrix $\boldsymbol{U}$ is defined as

$$\mathcal{Y} = \mathcal{X} \times_p \boldsymbol{U} = \mathrm{fold}_p(\boldsymbol{U}\boldsymbol{X}_{(p)}). \tag{2}$$

Definition 2 (Tensor-tensor product [49]): The tensor-tensor product (t-product) $\mathcal{Z} = \mathcal{X} * \mathcal{Y}$ of $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and

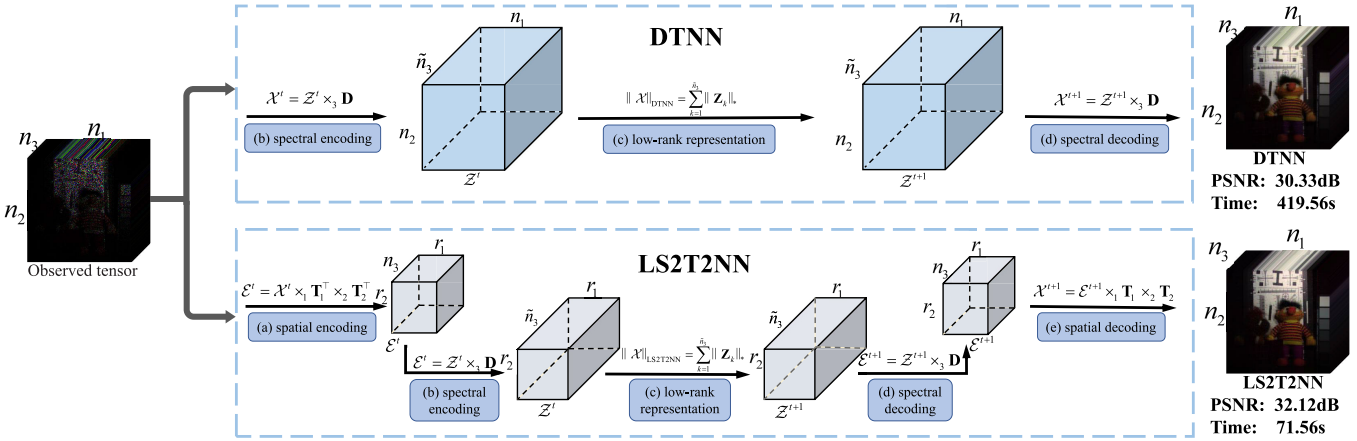


Fig. 1. Frameworks of DTNN and LS2T2NN. (a) The small-scale tensor $\mathcal{E}^t$ is obtained by applying learnable semi-orthogonal transforms to $\mathcal{X}^t$ along the spatial modes. (b) The spectral encoding, utilizing the learnable redundant transform, is performed on $\mathcal{E}^t$ along the spectral mode. (c) The low-rank representation operation is performed on each frontal slice of $\mathcal{Z}^t$. (d) The spectral decoding, utilizing the learnable redundant transform, is performed on $\mathcal{Z}^{t+1}$ along the spectral mode. (e) The spatial decoding, utilizing the learnable semi-orthogonal transforms, is performed on $\mathcal{E}^{t+1}$ along the spatial modes.

$\mathcal{Y} \in \mathbb{R}^{n_2 \times n_4 \times n_3}$ is a tensor of size $n_1 \times n_4 \times n_3$, whose (i, j)-th tube is given by

$$\mathcal{Z}(i, j, :) = \sum_{k=1}^{n_2} \mathcal{X}(i, k, :) \star \mathcal{Y}(k, j, :), \tag{3}$$

where $\star$ denotes the circular convolution operation of two tubes.

Definition 3 (Identity tensor, orthogonal tensor, and f-diagonal tensor [49]): The identity tensor $\mathcal{I} \in \mathbb{R}^{n \times n \times n_3}$ is that the first frontal slice is the identity matrix and other frontal slices are all zeros. A tensor $\mathcal{Q} \in \mathbb{R}^{n \times n \times n_3}$ is orthogonal if and only if it satisfies $\mathcal{Q}^H * \mathcal{Q} = \mathcal{Q} * \mathcal{Q}^H = \mathcal{I}$, where $\mathcal{Q}^H$ is the conjugate transpose of $\mathcal{Q}$. If each frontal slice of the tensor is diagonal, then it is an f-diagonal tensor.

Definition 4 (t-SVD [23], [49]): For $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, it can be factored as

$$\mathcal{X} = \mathcal{U} * \mathcal{S} * \mathcal{V}^H, \tag{4}$$

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 a f-diagonal tensor.

Definition 5 (Tensor tubal-rank and tensor multi-rank [23], [24]): For $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the tubal rank of $\mathcal{X}$ denoted as $\text{rank}_t(\mathcal{X})$ is the number of nonzero tubes of $\mathcal{S}$ from the $\mathcal{X} = \mathcal{U} * \mathcal{S} * \mathcal{V}^H$, i.e.,

$$\text{rank}_t(\mathcal{X}) := \#\{i : \mathcal{S}(i, i, :) \neq 0\}. \tag{5}$$

The tensor multi-rank of $\mathcal{X}$ denoted as $\text{rank}_m(\mathcal{X})$ is a vector whose k-th element is the rank of k-th frontal slice of $\hat{\mathcal{X}}$, where $\hat{\mathcal{X}} = \mathcal{X} \times_3 \boldsymbol{F}$ and $\boldsymbol{F}$ is the DFT matrix.

Definition 6 (TNN [25], [26]): For $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the tensor nuclear norm (TNN) of $\mathcal{X}$ is denoted by

$$\|\mathcal{X}\|_{\text{TNN}} \triangleq \frac{1}{n_3} \sum_{k=1}^{n_3} \|\boldsymbol{Z}_k\|_*, \tag{6}$$

where $\mathcal{X} = \mathcal{Z} \times_3 \boldsymbol{F}^\top$ and $\boldsymbol{Z}_k$ is the k-th frontal slice of $\mathcal{Z}$.

Definition 7 (DTNN [39]): For $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the dictionary-based TNN (DTNN) of $\mathcal{X}$ is denoted by

$$\|\mathcal{X}\|_{\text{DTNN}} \triangleq \sum_{k=1}^{\tilde{n}_3} \|\boldsymbol{Z}_k\|_*, \tag{7}$$

where $\mathcal{X} = \mathcal{Z} \times_3 \boldsymbol{D}$ and $\|\boldsymbol{D}(:, k)\|_2 = 1, k = 1, \cdots, \tilde{n}_3$, $\tilde{n}_3$ is number of columns of $\boldsymbol{D}$.

III. Model and Algorithm

In this section, we first propose the LS2T2NN model and discuss the relationship between LS2T2NN and DTNN with theoretical analysis. We then design the corresponding solution algorithm. Finally, we discuss the convergence and computational complexity of the algorithm.

A. Proposed LS2T2NN Model

The redundant transform can enhance the low-rankness of the frontal slices of the transformed tensor and produces significant recovery results. However, the significant recovery results come at the cost of an increase in the number of frontal slices of the transformed tensor, which results in an expensive computational cost in SVD calculations with the t-SVD framework. To reduce the computational burden, some fast algorithms have emerged with cheaper SVD calculations by projecting the original tensor to a small-scale tensor. These fast algorithms have gained computational advantages at the cost of accuracy. It can be seen from Fig. 2 that the recovery results by both DTNN and random t-SVD (RTSVD) methods are presented on video data *Flight*. These two methods are typical representatives of the transform-based TNN methods [34], [35], [36], [37], [38], [39], [40], [41], [42] and fast TNN-based methods [43], [44], [45], [46], [47]. The results indicate that DTNN provides superior restoration accuracy, but at the cost of higher computational complexity, while RTSVD offers lower computational cost but less impressive recovery result. In other words, the results show that DTNN faces computational limitations and that the performance of RTSVD is limited by stagnation.

TNN [26]	RTSVD [43]	DTNN [39]	LS2T2NN
PSNR: 26.55dB	PSNR: 26.61dB	PSNR: 27.63dB	PSNR: 29.99dB
Time: 26.42s	Time: 20.34s	Time: 579.16s	Time: 43.91s

Fig. 2. The visual results at the 10th frame, PSNR, and running time obtained by different methods on the video *Flight* under a sampling rate of 0.1.

To reduce computational cost without sacrificing recovery performance with the t-SVD framework, we first utilize learnable semi-orthogonal transforms as spatial projections to project the large-scale original tensor onto a small-scale intrinsic tensor. Next, we apply a learnable redundant transform along the spectral mode to the small-scale intrinsic tensor. Therefore, we model a tensor nuclear norm, called the learnable spatial-spectral transform-based tensor nuclear norm (LS2T2NN), which is defined as

$$\|\mathcal{X}\|_{\text{LS2T2NN}} \triangleq \sum_{k=1}^{\tilde{n}_3} \|\boldsymbol{Z}_k\|_*, \tag{8}$$

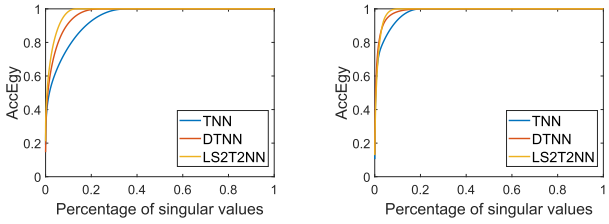
where $\mathcal{X} = \mathcal{Z} \times_1 \boldsymbol{T}_1 \times_2 \boldsymbol{T}_2 \times_3 \boldsymbol{D}$, $\boldsymbol{T}_1^\top \boldsymbol{T}_1 = \boldsymbol{I}_{r\times r}$, $\boldsymbol{T}_2^\top \boldsymbol{T}_2 = \boldsymbol{I}_{r\times r}$, and $\|\boldsymbol{D}(:,k)\|_2 = 1,\ k = 1,\ \cdots, \tilde{n}_3$. It is worth noting that $\boldsymbol{T}_1$, $\boldsymbol{T}_2$, and $\boldsymbol{D}$ are learnable at each iteration. Based on LS2T2NN, we design a corresponding LRTC model, which can be formulated mathematically as

$$\begin{aligned} \min_{\substack{\mathcal{X},\mathcal{Z},\\ \mathbf{D},\mathbf{T}_1,\mathbf{T}_2}} \ & \sum_{k=1}^{\tilde{n}_3} \|\boldsymbol{Z}_k\|_*,\ s.t.\ \mathcal{X} = \mathcal{Z} \times_1 \boldsymbol{T}_1 \times_2 \boldsymbol{T}_2 \times_3 \boldsymbol{D}, \\ & \boldsymbol{T}_1^\top \boldsymbol{T}_1 = \boldsymbol{I}_{r\times r}, \boldsymbol{T}_2^\top \boldsymbol{T}_2 = \boldsymbol{I}_{r\times r}, \\ & \|\boldsymbol{D}(:,k)\|_2 = 1,\ k = 1,\ \cdots, \tilde{n}_3, \mathcal{X}_\Omega = \mathcal{O}_\Omega, \end{aligned} \tag{9}$$

where $\boldsymbol{T}_1 \in \mathbb{R}^{n_1\times r}$ and $\boldsymbol{T}_2 \in \mathbb{R}^{n_2\times r}$ are learnable semi-orthogonal transforms, and $\boldsymbol{D} \in \mathbb{R}^{n_3\times \tilde{n}_3}$ is the learnable redundant transform, and Ω is the index set of observed entries in $\mathcal{O}$.

The advantages of our model are two-fold. First, our model can boost the low-rank representation. Second, our model can help reduce the computational cost by transforming the problem from size $n_1 \times n_2$ to $r \times r$, where $r \ll \min(n_1, n_2)$. We demonstrate the first advantage through theoretical and numerical analysis. From a theoretical perspective, given an underlying tensor $\mathcal{X} \in \mathbb{R}^{n_1\times\cdots\times n_N}$ and $\boldsymbol{T}_i \in \mathbb{R}^{n_i\times r_i}$, we have $\text{rank}(\mathcal{X} \times_1 \boldsymbol{T}_1^\top \cdots \times_N \boldsymbol{T}_N^\top) \le \text{rank}(\mathcal{X})$ [50]. This inequality proves that semi-orthogonal transforms can boost the low-rank representation. Additionally, Fig. 3 illustrates the accumulation energy ratio (AccEgy) [51] with the corresponding percentage of singular values of recovery results by TNN, DTNN, and LS2T2NN on video and MSI. This numerical analysis emphasizes that LS2T2NN results in a more concentrated energy distribution of singular values compared with TNN and DTNN, indicating its superior low-rank representation.

Remark 1: To reduce the computational cost, our strategy is from a model perspective, which differs from the random algorithm [43] that is a classic representation from an algorithmic perspective. Thus, our projection



(a) Video *Rhino* (b) MSI *Pompom*

Fig. 3. The AccEgy with the corresponding percentage of singular values of recovered results by TNN, DTNN, and LS2T2NN.

is a data-dependent learnable semi-orthogonal transform. Unlike the data-independent random projection, the learnable semi-orthogonal transform is embedded in the model, which is updated by (34) or (36) at each iteration.

Next, an interesting question is what is the relationship between LS2T2NN and DTNN? For a given redundant transform $\boldsymbol{D}$, we can find a relationship between LS2T2NN and DTNN, i.e.,

$$\|\mathcal{X}\|_{\text{LS2T2NN}} \le \|\mathcal{X}\|_{\text{DTNN}}. \tag{10}$$

Before discussing this relationship, we introduce a lemma.

Lemma 1: (Von Neuman's Trace Inequality [52]). If matrices $\boldsymbol{M}$ and $\boldsymbol{N}$ with size $m \times n$ have singular values $\sigma_1(\boldsymbol{M}) \ge \sigma_2(\boldsymbol{M}) \ge \cdots \ge \sigma_r(\boldsymbol{M})$, and $\sigma_1(\boldsymbol{N}) \ge \sigma_2(\boldsymbol{N}) \ge \cdots \ge \sigma_r(\boldsymbol{N})$, respectively, where $r = \min(m, n)$. Then, we have

$$\text{Tr}\left(\boldsymbol{M}^\top \boldsymbol{N}\right) \le \sum_{i=1}^{r} \sigma_i(\boldsymbol{M})\, \sigma_i(\boldsymbol{N}).$$

Specially, if $\boldsymbol{M}$ is a semi-orthogonal matrix, we can get

$$\text{Tr}\left(\boldsymbol{M}^\top \boldsymbol{N}\right) \le \sum_{i=1}^{r} \sigma_i(\boldsymbol{M})\, \sigma_i(\boldsymbol{N}) = \sum_{i=1}^{r} \sigma_i(\boldsymbol{N}) = \|\boldsymbol{N}\|_*.$$

Based on Lemma 1, we have the following Theorem.

Theorem 1: For a given redundant transform $\boldsymbol{D} \in \mathbb{R}^{n_3\times\tilde{n}_3}$, semi-orthogonal matrices $\boldsymbol{T}_1 \in \mathbb{R}^{n_1\times r}$ and $\boldsymbol{T}_2 \in \mathbb{R}^{n_2\times r}$, $\mathcal{A} \in \mathbb{R}^{n_1\times n_2\times\tilde{n}_3}$, $\mathcal{B} \in \mathbb{R}^{r\times r\times\tilde{n}_3}$ and $\mathcal{X} \in \mathbb{R}^{n_1\times n_2\times n_3}$. They satisfy $\mathcal{X} = \mathcal{A} \times_3 \boldsymbol{D} = \mathcal{B} \times_1 \boldsymbol{T}_1 \times_2 \boldsymbol{T}_2 \times_3 \boldsymbol{D}$, such that

$$\|\mathcal{X}\|_{\text{LS2T2NN}} \le \|\mathcal{X}\|_{\text{DTNN}}, \tag{11}$$

where $\|\mathcal{X}\|_{\text{LS2T2NN}} = \sum_{k=1}^{\tilde{n}_3} \|\boldsymbol{B}_k\|_*$, $\|\mathcal{X}\|_{\text{DTNN}} = \sum_{k=1}^{\tilde{n}_3} \|\boldsymbol{A}_k\|_*$. $\boldsymbol{A}_k$ and $\boldsymbol{B}_k$ are the k-th frontal slices of $\mathcal{A}$ and $\mathcal{B}$, respectively.

Proof: Since the redundancy dictionary $\boldsymbol{D}$ is the same in DTNN and LS2T2NN, we have $\mathcal{A} = \mathcal{B} \times_1 \boldsymbol{T}_1 \times_2 \boldsymbol{T}_2$. To prove inequality (11), we need the following inequality

$$\|\boldsymbol{T}^\top \boldsymbol{A}\|_* \le \|\boldsymbol{A}\|_*, \tag{12}$$

where $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ is any matrix and $\boldsymbol{T} \in \mathbb{R}^{n\times r}$ is a semi-orthogonal matrix which satisfies $\boldsymbol{T}^\top \boldsymbol{T} = \boldsymbol{I}_{r\times r}$. Let $\boldsymbol{B} = \boldsymbol{T}^\top \boldsymbol{A}$, where $\boldsymbol{B} \in \mathbb{R}^{r\times n}$. Based on SVD, $\boldsymbol{A}$ and $\boldsymbol{B}$ can be rewritten as follows:

$$\boldsymbol{A} = \boldsymbol{L}_{\boldsymbol{A}} \Sigma_{\boldsymbol{A}} \boldsymbol{R}_{\boldsymbol{A}}^\top, \boldsymbol{B} = \boldsymbol{L}_{\boldsymbol{B}} \Sigma_{\boldsymbol{B}} \boldsymbol{R}_{\boldsymbol{B}}^\top,$$

where $\boldsymbol{L}_{\boldsymbol{A}}$, $\boldsymbol{R}_{\boldsymbol{A}}$, $\boldsymbol{L}_{\boldsymbol{B}}$, and $\boldsymbol{R}_{\boldsymbol{B}}$ are orthogonal matrices, $\Sigma_{\boldsymbol{A}}$, and $\Sigma_{\boldsymbol{B}}$ are diagonal matrices. Thus, we can get $\Sigma_{\boldsymbol{B}} = \boldsymbol{L}_{\boldsymbol{B}}^\top \boldsymbol{B} \boldsymbol{R}_{\boldsymbol{B}}$.

Due to $\|\boldsymbol{B}\|_* = \mathrm{Tr}(\Sigma_{\boldsymbol{B}}) = \mathrm{Tr}(\boldsymbol{L}_{\boldsymbol{B}}^\top \boldsymbol{B}\boldsymbol{R}_{\boldsymbol{B}}) = \mathrm{Tr}(\boldsymbol{L}_{\boldsymbol{B}}^\top \boldsymbol{T}^\top \boldsymbol{A}\boldsymbol{R}_{\boldsymbol{B}})$, we can perceive that $\boldsymbol{L}_{\boldsymbol{B}}^\top \boldsymbol{T}^\top$ is a semi-orthogonal matrix, which satisfies $(\boldsymbol{L}_{\boldsymbol{B}}^\top \boldsymbol{T}^\top)(\boldsymbol{L}_{\boldsymbol{B}}^\top \boldsymbol{T}^\top)^\top = \boldsymbol{L}_{\boldsymbol{B}}^\top \boldsymbol{T}^\top \boldsymbol{T}\boldsymbol{L}_{\boldsymbol{B}} = \boldsymbol{I}_{r\times r}$. Base on Lemma 1, we have

$$\|\boldsymbol{B}\|_* = \mathrm{Tr}(\boldsymbol{L}_{\boldsymbol{B}}^\top \boldsymbol{T}^\top \boldsymbol{A}\boldsymbol{R}_{\boldsymbol{B}}) \le \|\boldsymbol{A}\boldsymbol{R}_{\boldsymbol{B}}\|_*.$$

Since $\boldsymbol{R}_{\boldsymbol{A}}^\top$ and $\boldsymbol{R}_{\boldsymbol{B}}$ are orthogonal, $\boldsymbol{R}_{\boldsymbol{A}}^\top \boldsymbol{R}_{\boldsymbol{B}}$ is orthogonal. The SVD of $\boldsymbol{A}\boldsymbol{R}_{\boldsymbol{B}}$ is $\boldsymbol{A}\boldsymbol{R}_{\boldsymbol{B}} = \boldsymbol{L}_{\boldsymbol{A}}\Sigma_{\boldsymbol{A}}\boldsymbol{R}_{\boldsymbol{A}}^\top \boldsymbol{R}_{\boldsymbol{B}}$. We have $\|\boldsymbol{A}\boldsymbol{R}_{\boldsymbol{B}}\|_* = \mathrm{Tr}(\Sigma_{\boldsymbol{A}}) = \|\boldsymbol{A}\|_*$. Therefore, we have

$$\|\boldsymbol{T}^\top \boldsymbol{A}\|_* = \|\boldsymbol{B}\|_* \le \|\boldsymbol{A}\boldsymbol{R}_{\boldsymbol{B}}\|_* = \|\boldsymbol{A}\|_*.$$

For $\mathcal{A} = \mathcal{B} \times_1 \boldsymbol{T}_1 \times_2 \boldsymbol{T}_2$, we have

$$\boldsymbol{A}_{(1)} = \boldsymbol{T}_1 \boldsymbol{B}_{(1)} \boldsymbol{T}_2^\top \Rightarrow \boldsymbol{T}_1^\top \boldsymbol{A}_{(1)} \boldsymbol{T}_2 = \boldsymbol{B}_{(1)},$$

where $\boldsymbol{A}_{(1)}$ and $\boldsymbol{B}_{(1)}$ are the mode-1 unfolding matrix of $\mathcal{A}$ and $\mathcal{B}$, respectively. Due to the definition of unfolding operator, we can obtain

$$\begin{aligned} \boldsymbol{T}_1^\top \left[\boldsymbol{A}_1, \ \cdots, \boldsymbol{A}_{\tilde{n}_3}\right] \boldsymbol{T}_2 &= \left[\boldsymbol{B}_1, \cdots, \boldsymbol{B}_{\tilde{n}_3}\right], \\ \left[\boldsymbol{T}_1^\top \boldsymbol{A}_1 \boldsymbol{T}_2, \cdots, \boldsymbol{T}_1^\top \boldsymbol{A}_{\tilde{n}_3} \boldsymbol{T}_2\right] &= \left[\boldsymbol{B}_1, \cdots, \boldsymbol{B}_{\tilde{n}_3}\right], \end{aligned}$$

where $\boldsymbol{A}_k$ and $\boldsymbol{B}_k$ are the k-th slices of $\mathcal{A}$ and $\mathcal{B}$, respectively. Based on inequation (12), we can obtain

$$\begin{aligned} \|\boldsymbol{B}_k\|_* &= \|\boldsymbol{T}_1^\top \boldsymbol{A}_k \boldsymbol{T}_2\|_* \le \|\boldsymbol{A}_k \boldsymbol{T}_2\|_* \\ &= \|\boldsymbol{T}_2^\top \boldsymbol{A}_k^\top\|_* \le \|\boldsymbol{A}_k^\top\|_* = \|\boldsymbol{A}_k\|_*, \end{aligned}$$

where $k \in \{1, \cdots, \tilde{n}_3\}$. Based on the definitions of DTNN and LS2T2NN, we have

$$\|\mathcal{X}\|_{\mathrm{LS2T2NN}} = \sum_{k=1}^{\tilde{n}_3} \|\boldsymbol{B}_k\|_* \le \sum_{k=1}^{\tilde{n}_3} \|\boldsymbol{A}_k\|_* = \|\mathcal{X}\|_{\mathrm{DTNN}}.$$

□

B. Proposed Algorithm

To solve the proposed model, we introduce the auxiliary variable $\mathcal{E} = \mathcal{Z} \times_3 \boldsymbol{D}$. So, the optimization problem (9) can be equivalently rewritten as follows:

$$\begin{aligned} \min_{\substack{\mathcal{X},\mathcal{E},\mathcal{Z},\\ \mathbf{D},\mathbf{T}_1,\mathbf{T}_2}} & \sum_{k=1}^{\tilde{n}_3} \|\boldsymbol{Z}_k\|_*, \ s.t. \ \mathcal{X} = \mathcal{E} \times_1 \boldsymbol{T}_1 \times_2 \boldsymbol{T}_2, \\ & \mathcal{E} = \mathcal{Z} \times_3 \boldsymbol{D}, \boldsymbol{T}_1^\top \boldsymbol{T}_1 = \boldsymbol{I}_{r\times r}, \boldsymbol{T}_2^\top \boldsymbol{T}_2 = \boldsymbol{I}_{r\times r}, \\ & \|\boldsymbol{D}(:,k)\|_2 = 1, \ k = 1, \ \cdots, \tilde{n}_3, \mathcal{X}_\Omega = \mathcal{O}_\Omega, \end{aligned} \tag{13}$$

where Ω is the index set of observed entries in $\mathcal{O}$. Moreover, we introduce three indicator functions $\Phi(\mathcal{X})$, $\Psi(\boldsymbol{T})$, and $\Upsilon(\boldsymbol{D})$, i.e.,

$$\Phi(\mathcal{X}) = \begin{cases} 0, & \mathcal{X}_\Omega = \mathcal{O}_\Omega, \\ +\infty, & otherwise, \end{cases} \tag{14}$$

$$\Psi(\boldsymbol{T}) = \begin{cases} 0, & \boldsymbol{T}^\top \boldsymbol{T} = \boldsymbol{I}_{r\times r}, \\ +\infty, & otherwise, \end{cases} \tag{15}$$

$$\Upsilon(\boldsymbol{D}) = \begin{cases} 0, & \|\boldsymbol{D}(:,k)\|_2 = 1, \\ +\infty, & otherwise. \end{cases} \tag{16}$$

As directly solving the optimization problem in (13) is difficult, we reformulate problem (13) as the unconstrained problem via the half quadratic splitting (HQS) [53], [54] technique, that is

$$\begin{aligned} \min_{\substack{\mathcal{X},\mathcal{E},\mathcal{Z},\\ \mathbf{D},\mathbf{T}_1,\mathbf{T}_2}} & \sum_{k=1}^{\tilde{n}_3} \|\boldsymbol{Z}_k\|_* + \frac{\alpha}{2}\|\mathcal{X} - \mathcal{E} \times_1 \boldsymbol{T}_1 \times_2 \boldsymbol{T}_2\|_F^2 \\ & + \frac{\beta}{2}\|\mathcal{E} - \mathcal{Z} \times_3 \boldsymbol{D}\|_F^2 + \Phi(\mathcal{X}) + \Psi(\boldsymbol{T}_1) \\ & + \Psi(\boldsymbol{T}_2) + \Upsilon(\boldsymbol{D}), \end{aligned} \tag{17}$$

where α and β are penalty parameters. Since the optimization problem (17) is nonconvex and has six variables, we apply the PAM [55] to design an efficient algorithm with the convergence guarantee. In each iteration, $\mathcal{X}$, $\mathcal{E}$, $\{\mathcal{Z}, \boldsymbol{D}\}$, and $\boldsymbol{T}_i (i \in \{1, 2\})$ are alternately updated as

$$\begin{aligned} \mathcal{X}^{t+1} &= \arg\min_{\mathcal{X}} \{L(\mathcal{X}, \mathcal{E}^t, \mathcal{Z}^t, \boldsymbol{D}^t, \boldsymbol{T}_i^t) \\ &\quad + \frac{\rho_1}{2}\|\mathcal{X} - \mathcal{X}^t\|_F^2\}, \\ \mathcal{E}^{t+1} &= \arg\min_{\mathcal{E}} \{L\left(\mathcal{X}^{t+1}, \mathcal{E}, \mathcal{Z}^t, \boldsymbol{D}^t, \boldsymbol{T}_i^t\right) \\ &\quad + \frac{\rho_2}{2}\|\mathcal{E} - \mathcal{E}^t\|_F^2\}, \\ \{\mathcal{Z}^{t+1}, \boldsymbol{D}^{t+1}\} &= \arg\min_{\mathcal{Z},\boldsymbol{D}} \{L\left(\mathcal{X}^{t+1}, \mathcal{E}^{t+1}, \mathcal{Z}, \boldsymbol{D}^t, \boldsymbol{T}_i^t\right) \\ &\quad + \frac{\rho_3}{2}\|\mathcal{Z} - \mathcal{Z}^t\|_F^2 + \frac{\rho_4}{2}\|\boldsymbol{D} - \boldsymbol{D}^t\|_F^2\}, \\ \boldsymbol{T}_i^{t+1} &= \arg\min_{\boldsymbol{T}_i} \{L\left(\mathcal{X}^{t+1}, \mathcal{E}^{t+1}, \mathcal{Z}^{t+1}, \boldsymbol{D}^{t+1}, \boldsymbol{T}_i\right) \\ &\quad + \frac{\rho_5}{2}\|\boldsymbol{T}_i - \boldsymbol{T}_i^t\|_F^2\}, \end{aligned} \tag{18}$$

where $L(\mathcal{X}, \mathcal{E}, \mathcal{Z}, \boldsymbol{D}, \boldsymbol{T}_i)$ is the objective function in (17). Next, we present each subproblem in detail.

- The $\mathcal{X}$ subproblem is as follows:

$$\begin{aligned} \mathcal{X}^{t+1} &= \arg\min_{\mathcal{X}} \frac{\alpha}{2}\|\mathcal{X} - \mathcal{E}^t \times_1 \boldsymbol{T}_1^t \times_2 \boldsymbol{T}_2^t\|_F^2 \\ &\quad + \frac{\rho_1}{2}\|\mathcal{X} - \mathcal{X}^t\|_F^2 + \Phi(\mathcal{X}). \end{aligned} \tag{19}$$

  The closed-form solution of (19) is

$$\mathcal{X}^{t+1} = \left(\frac{\alpha\left(\mathcal{E}^t \times_1 \boldsymbol{T}_1^t \times_2 \boldsymbol{T}_2^t\right) + \rho_1 \mathcal{X}^t}{\alpha + \rho_1}\right)_{\Omega^C} + \mathcal{O}_\Omega, \tag{20}$$

  where Ω^C is the complementary set of Ω.
- The $\mathcal{E}$ subproblem is as follows:

$$\begin{aligned} \mathcal{E}^{t+1} = \arg\min_{\mathcal{E}} & \frac{\alpha}{2}\|\mathcal{X}^{t+1} - \mathcal{E} \times_1 \boldsymbol{T}_1^t \times_2 \boldsymbol{T}_1^t\|_F^2 \\ & + \frac{\beta}{2}\|\mathcal{E} - \mathcal{Z}^t \times_3 \boldsymbol{D}^t\|_F^2 + \frac{\rho_2}{2}\|\mathcal{E} - \mathcal{E}^t\|_F^2. \end{aligned} \tag{21}$$

The closed-form solution of (21) is

$$\mathcal{E}^{t+1} = \left(\frac{\alpha \mathcal{H}^{t+1} + \beta \left(\mathcal{Z}^t \times_3 \boldsymbol{D}^t \right) + \rho_1 \mathcal{E}^t}{\alpha + \beta + \rho_1} \right), \tag{22}$$

where $\mathcal{H}^{t+1} = \mathcal{X}^{t+1} \times_1 \boldsymbol{T}_1^{\top t} \times_2 \boldsymbol{T}_2^{\top t}$.

- The $\{\mathcal{Z}, \boldsymbol{D}\}$ subproblem is as follows:

$$\begin{aligned} \{\mathcal{Z}^{t+1}, \boldsymbol{D}^{t+1}\} = \arg\min_{\mathcal{Z}, \boldsymbol{D}} & \sum_{k=1}^{\tilde{n}_3} \|\boldsymbol{Z}_k\|_* + \frac{\rho_3}{2} \|\mathcal{Z} - \mathcal{Z}^t\|_F^2 \\ & + \frac{\beta}{2} \|\mathcal{E}^{t+1} - \mathcal{Z} \times_3 \boldsymbol{D}\|_F^2, + \frac{\rho_4}{2} \|\boldsymbol{D} \\ & - \boldsymbol{D}^t\|_F^2 + \Upsilon(\boldsymbol{D}). \end{aligned} \tag{23}$$

Since $\mathcal{Z}$ and $\boldsymbol{D}$ are coupled with each other in the optimization problem, we adopt an alternating update strategy [39], [56] to solve them. Then, problem (23) can be decomposed into solving $\tilde{n}_3$ subproblems of $\mathcal{Z}$ and $\boldsymbol{D}$, which are equivalent to

$$\begin{cases} \arg\min_{\boldsymbol{Z}_k} \sum_{k=1}^{\tilde{n}_3} \|\boldsymbol{Z}_k\|_* + \frac{\rho_3}{2} \sum_{k=1}^{\tilde{n}_3} \left\| \boldsymbol{Z}_k - \boldsymbol{Z}_k^t \right\|_F^2 \\ + \frac{\beta}{2} \left\| \boldsymbol{E}_{(3)}^{t+1} - \sum_{k=1}^{\tilde{n}_3} \boldsymbol{d}_k \boldsymbol{z}_k^\top \right\|_F^2, \\ \arg\min_{\boldsymbol{d}_k} \frac{\beta}{2} \left\| \boldsymbol{E}_{(3)}^{t+1} - \sum_{k=1}^{\tilde{n}_3} \boldsymbol{d}_k \boldsymbol{z}_k^\top \right\|_F^2 + \Upsilon(\boldsymbol{D}) \\ + \frac{\rho_4}{2} \sum_{k=1}^{\tilde{n}_3} \left\| \boldsymbol{d}_k - \boldsymbol{d}_k^t \right\|_F^2, \end{cases} \tag{24}$$

where $\boldsymbol{Z}_{(3)}$ and $\boldsymbol{E}_{(3)}$ are the mode-3 unfolding of $\mathcal{Z}$ and $\mathcal{E}$, respectively. $\boldsymbol{Z}_k = \mathcal{Z}(:, :, k)$ is the k-th frontal slice of $\mathcal{Z}$, $\boldsymbol{d}_k = \boldsymbol{D}(:, k)$ is the k-th atom of $\boldsymbol{D}$, and

$$\begin{aligned} \boldsymbol{Z}_{(3)} &= \left[\boldsymbol{z}_1, \cdots, \boldsymbol{z}_k, \cdots, \boldsymbol{z}_{\tilde{n}_3} \right]^\top \\ &= \left[\text{vec}\left(\boldsymbol{Z}_1\right), \cdots, \text{vec}\left(\boldsymbol{Z}_k\right), \cdots, \text{vec}\left(\boldsymbol{Z}_{\tilde{n}_3}\right) \right]^\top, \end{aligned} \tag{25}$$

$$\boldsymbol{D} = \left[\boldsymbol{d}_1, \cdots, \boldsymbol{d}_k, \cdots, \boldsymbol{d}_{\tilde{n}_3} \right]. \tag{26}$$

For the sake of simplicity, we define

$$\begin{cases} \widehat{\boldsymbol{Z}}_k^t = \left[\boldsymbol{z}_1^{t+1}, \cdots, \boldsymbol{z}_{k-1}^{t+1}, \boldsymbol{z}_{k+1}^t, \cdots, \boldsymbol{z}_{\tilde{n}_3}^t \right]^\top, \\ \widehat{\boldsymbol{D}}_k^t = \left[\boldsymbol{d}_1^{t+1}, \cdots, \boldsymbol{d}_{k-1}^{t+1}, \boldsymbol{d}_{k+1}^t, \ldots, \boldsymbol{d}_{\tilde{n}_3}^t \right], \\ \boldsymbol{R}_k^t = \mathbf{E}_{(3)}^t - \widehat{\boldsymbol{D}}_k^t \widehat{\boldsymbol{Z}}_k^t. \end{cases} \tag{27}$$

Based on (27), the $\boldsymbol{Z}_k$-subproblem in (24) is equivalent to

$$\min_{\boldsymbol{Z}_k} \|\boldsymbol{Z}_k\|_* + \frac{\beta}{2} \left\| \boldsymbol{R}_k^t - \boldsymbol{d}_k^t \boldsymbol{z}_k^\top \right\|_F^2 + \frac{\rho_3}{2} \left\| \boldsymbol{Z}_k - \boldsymbol{Z}_k^t \right\|_F^2. \tag{28}$$

The closed-form solution of (28) is

$$\boldsymbol{Z}_k^{t+1} = \text{SVT}_{\frac{1}{\beta+\rho_3}} \left(\frac{\beta \, \text{vec}^{-1}\left(\boldsymbol{R}_k^{t\top} d_k^t \right) + \rho_3 \boldsymbol{Z}_k^t}{\beta + \rho_3} \right), \tag{29}$$

where $\text{SVT}_\tau(\cdot)$ is singular value thresholding (SVT) operator [57]. Similarly, the $\boldsymbol{d}_k$-subproblem in (24) is equivalent to

$$\min_{\boldsymbol{d}_k} \frac{\beta}{2} \left\| \boldsymbol{R}_k^t - \boldsymbol{d}_k \boldsymbol{z}_k^{t+1\top} \right\|_F^2 + \frac{\rho_4}{2} \left\| \boldsymbol{d}_k - \boldsymbol{d}_k^t \right\|_F^2 + \Upsilon(\boldsymbol{D}). \tag{30}$$

The closed-form solution of (30) is

$$\boldsymbol{d}_k^{t+1} = \frac{\beta \boldsymbol{R}_k^t \, \text{vec}\left(\boldsymbol{Z}_k^{t+1} \right) + \rho_4 \boldsymbol{d}_k^t}{\left\| \beta \boldsymbol{R}_k^t \, \text{vec}\left(\boldsymbol{Z}_k^{t+1} \right) + \rho_4 \boldsymbol{d}_k^t \right\|_2}. \tag{31}$$

- The $\boldsymbol{T}_i (i \in \{1, 2\})$ subproblems are as follows:

$$\begin{aligned} \boldsymbol{T}_i^{t+1} = \arg\min_{\boldsymbol{T}_i} & \frac{\alpha}{2} \|\mathcal{X} - \mathcal{E} \times_1 \boldsymbol{T}_1 \times_2 \boldsymbol{T}_2\|_F^2 \\ & + \frac{\rho_5}{2} \|\boldsymbol{T}_i - \boldsymbol{T}_i^t\|_F^2 + \Psi(\boldsymbol{T}_i). \end{aligned} \tag{32}$$

1) Update $\boldsymbol{T}_1$ subproblem:

$$\boldsymbol{T}_1^{t+1} = \arg\max_{\boldsymbol{T}_1} \text{Tr}\left(\boldsymbol{Q}_1^t \boldsymbol{T}_1 \right) - \Psi(\boldsymbol{T}_1), \tag{33}$$

where $\boldsymbol{Q}_1^t = \left(\alpha \boldsymbol{X}_{(1)}^{t+1} \left(\mathcal{E} \times_2 \boldsymbol{T}_2^{t+1} \right)_{(1)}^\top + \rho_5 \left(\boldsymbol{T}_1^t \right)^\top \right)$, $\text{Tr}(\boldsymbol{X})$ denotes the trace of matrix $\boldsymbol{X}$. This is an orthogonal procrustes problem [58], and the closed-form solution of (33) is

$$\boldsymbol{T}_1 = \boldsymbol{V}_1^{t+1} (\boldsymbol{U}_1^{t+1})^\top, \tag{34}$$

where $\boldsymbol{U}_1^{t+1}$, $\boldsymbol{S}_1^{t+1}$, and $\boldsymbol{V}_1^{t+1}$ are results of SVD on $\boldsymbol{Q}_1^t$.

2) Update $\boldsymbol{T}_2$ subproblem:

$$\boldsymbol{T}_2^{t+1} = \arg\max_{\boldsymbol{T}_2} \text{Tr}\left(\boldsymbol{Q}_2^t \boldsymbol{T}_2 \right) - \Psi(\boldsymbol{T}_2), \tag{35}$$

where $\boldsymbol{Q}_2^t = \left(\alpha \boldsymbol{X}_{(2)}^{t+1} \left(\mathcal{E} \times_1 \boldsymbol{T}_1^{t+1} \right)_{(2)}^\top + \rho_5 \left(\boldsymbol{T}_2^t \right)^\top \right)$. the closed-form solution of (35) is

$$\boldsymbol{T}_2 = \boldsymbol{V}_2^{t+1} (\boldsymbol{U}_2^{t+1})^\top, \tag{36}$$

where $\boldsymbol{U}_2^{t+1}$, $\boldsymbol{S}_2^{t+1}$, and $\boldsymbol{V}_2^{t+1}$ are results of SVD on $\boldsymbol{Q}_2^t$.

Finally, we summarize the PAM-based solving algorithm for the LS2T2NN model in Algorithm 1.

C. Convergence Analysis

In this section, we provide the theoretical convergence of our algorithm. For convenience, we denote the objective function of problem (17) as $L\left(\mathcal{X}^t, \mathcal{E}^t, \mathcal{Z}^t, \boldsymbol{D}^t, \boldsymbol{T}_i^t \right)$. Then, we have the following theoretical guarantee:

Theorem 2: The sequence $\left(\mathcal{X}^t, \mathcal{E}^t, \mathcal{Z}^t, \boldsymbol{D}^t, \boldsymbol{T}_i^t \right)_{t \in \mathbb{N}}$ generated by Algorithm 1 converges to a critical point of L.

The proof of Theorem 2 is presented in the supplementary material.

Algorithm 1 PAM-Based Algorithm for LS2T2NN.

Input: The observed tensor $\mathcal{O} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the index Ω, the estimated rank along spatial modes $r = r_1 = r_2$, parameter α and β, proximal parameters $\rho = \rho_1 = \rho_2 = \rho_3 = \rho_4 = \rho_5$, and the maximum iteration $t_{max} = 1000$.

Output: The reconstructed tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$.

Initialization: The iteration $t = 0$, $\mathcal{X}^0$, $\mathcal{E}^0$, $\mathcal{Z}^0$, $\boldsymbol{D}^0$, $\boldsymbol{T}_1^0$, and $\boldsymbol{T}_2^0$.
while not converged and $t < t_{max}$ **do**
  Update $\mathcal{X}^{t+1}$ via Eq. (20);
  Update $\mathcal{E}^{t+1}$ via Eq. (22);
  Update $\mathcal{Z}^{t+1}$ via Eq. (29);
  Update $\boldsymbol{D}^{t+1}$ via Eq. (31);
  Update $\boldsymbol{T}_1^{t+1}$ via Eq. (34);
  Update $\boldsymbol{T}_2^{t+1}$ via Eq. (36);
  Check the convergence conditions

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

end while

TABLE II
THE COMPUTATIONAL COMPLEXITY FOR DIFFERENT METHODS

Method	Computational complexity
TNN	$\mathcal{O}(n^2 n_3 \log n_3 + n^3 n_3)$
UTNN	$\mathcal{O}(n^2 n_3^2 + n^3 n_3)$
DTNN	$\mathcal{O}((n_3\tilde{n}_3 + n_3 + n)n^2\tilde{n}_3)$
LS2T2NN	$\mathcal{O}((r+n)rnn_3 + (n_3\tilde{n}_3 + n_3 + r)r^2\tilde{n}_3 + r^2 n)$

D. Computational Complexity Analysis

In this section, we analyze the computational complexity of PAM for LS2T2NN in Algorithm 1. For $\mathcal{X} \in \mathbb{R}^{n \times n \times n_3}$, we assume that the estimated rank along spatial mode is r and the size of the redundant transform is $n_3 \times \tilde{n}_3$ in our algorithm. The computational complexity at each iteration of PAM for LS2T2NN can be concluded by updating $\mathcal{X}$, updating $\{\mathcal{Z}, \boldsymbol{D}\}$, updating $\mathcal{E}$, and updating $\boldsymbol{T}_i$, which cost $\mathcal{O}(rn^2 n_3 + r^2 n n_3)$, $\mathcal{O}(r^2\tilde{n}_3(n_3\tilde{n}_3 + n_3 + r))$, $\mathcal{O}(rn^2 n_3 + r^2 n n_3 + r^2 n_3 \tilde{n}_3)$, and $\mathcal{O}(r^2 n n_3 + r n^2 n_3 + r^2 n)$, respectively. Therefore, the computational complexity of PAM for LS2T2NN is $\mathcal{O}((r+n)rnn_3 + (n_3\tilde{n}_3 + n_3 + r)r^2\tilde{n}_3 + r^2 n)$. Table II lists the computational complexity of PAM for LS2T2NN and the baselines, such as TNN, UTNN, and DTNN, for reference. From Table II, the complexity of PAM for LS2T2NN is less than that of PAM for DTNN when $r \ll n$.

IV. NUMERICAL EXPERIMENTS

In this section, we conduct numerical experiments on videos, MSIs, and color images to examine the effectiveness of LS2T2NN for LRTC. Especially, we compare with other state-of-the-art methods, including a t-SVD baseline method TNN [32], a fast random t-SVD-based method RTSVD [43], a fast tensor QR method TQRTNN [28], a unitary transform-based method UTNN [38] and a dictionary encoding-based method DTNN [39]. All test data is linearly mapped to [0,1] range. For all methods, we use a stopping criterion where the relative change is $\left\| \mathcal{X}^{t+1} - \mathcal{X}^t \right\|_F / \left\| \mathcal{X}^t \right\|_F \leq 10^{-4}$. Since the proposed method is nonconvex, we utilize a linear interpolation strategy [59] to initialize $\mathcal{X}^0$, and the DCT matrix to initialize $\boldsymbol{T}_i^0$. For the compared methods, we manually adjust parameters to ensure optimal performance according to author's suggestions. For numerical comparison, we use the peak signal-to-noise rate (PSNR) [60] and the structural similarity (SSIM) [61] to evaluate the performance of different methods. All numerical experiments are implemented in Windows 10 64-bit and MATLAB R2022a on a desktop computer with an Intel(R) Core(TM) i9-12900 CPU at 2.40 GHz with 64GB memory of RAM.

A. Experiments on Videos

We evaluate the performance of LS2T2NN on five videos from the NTT database,[1] including *Rhino*, *Bird*, *Horse*, *Wildcat*, and *Flight*. The spatial size of each video is 288×352 . We select the first 30 frames of each video in our experiments. The size of resulting data is $288 \times 352 \times 30$. For each data, sampling rates (SRs) are set to be 0.10, 0.15, 0.20, and 0.30, respectively.

Table III presents the PSNR, SSIM, and running time obtained by different methods for different SRs. Although TNN, RTSVD, and TQRTNN are competitive in terms of running time, their performance is unsatisfactory in terms of PSNR and SSIM. The UTNN and DTNN perform better than the above three methods in terms of PSNR and SSIM but at the cost of longer running time. Our method, LS2T2NN, achieves superior performance in both PSNR and SSIM compared to other methods. Moreover, the proposed method outperforms DTNN in terms of running time, thanks to its spatial projections. These observations indicate that our algorithm is computationally efficient at each step. Fig. 4 illustrates the comparison of recovery results by TNN, RTSVD, TQRTNN, UTNN, DTNN, and LS2T2NN on videos with SR = 0.2. As shown in this figure, we observe that most of the competing methods only recover the main body of the latent data and produce blurred edges (see the zoom-in regions of *Bird* and *Horse*) or non-existent noise (see the zoom-in regions of *Wildcat* and *Flight*). The recovery results produced by the proposed method have better visual quality, especially in image edge and local image detail.

B. Experiments on MSIs

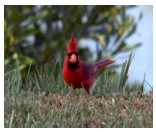
We test the effectiveness of the proposed method on five MSIs from the CAVE database[2]: *Balloon*, *Beer*, *Feather*, *Pompom*, and *Toy*. In our experiments, each MSI is resized to $256 \times 256 \times 31$. For each MSI, SRs are set to 0.05, 0.10, 0.15, and 0.20, respectively.

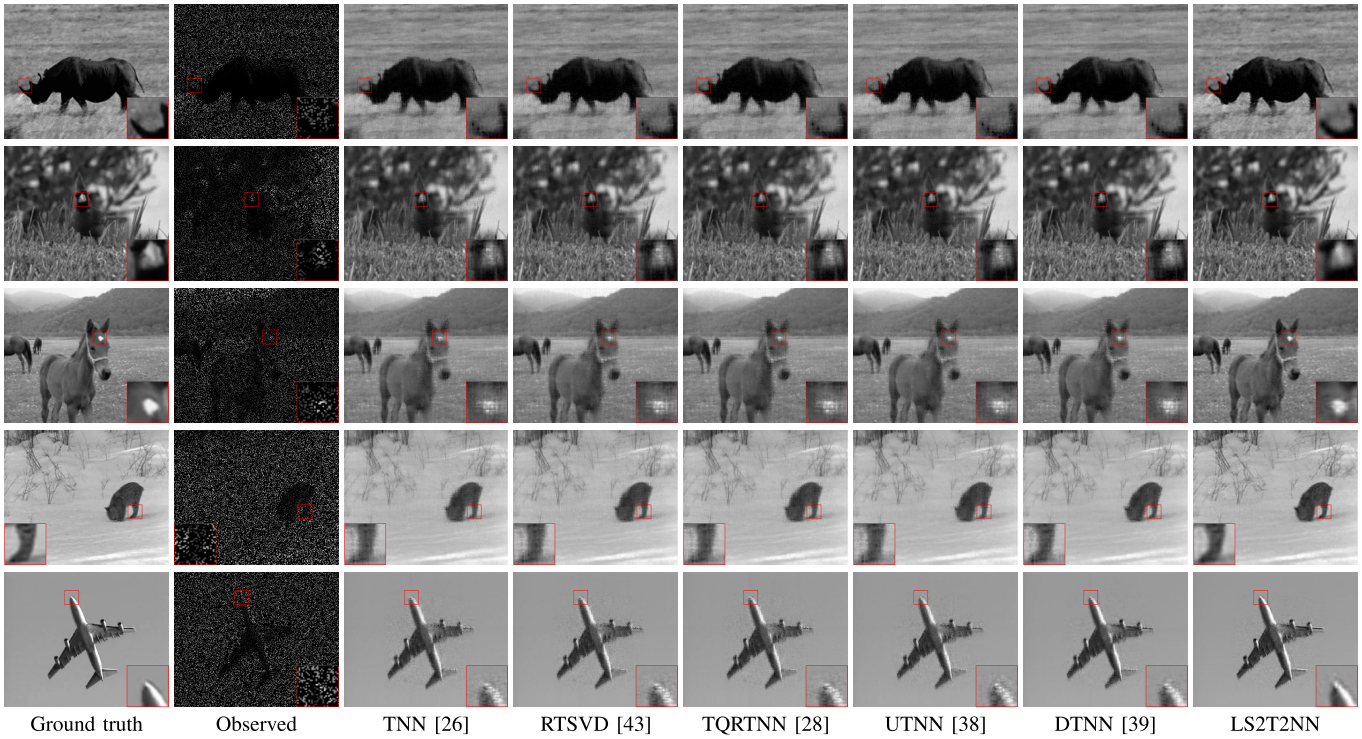
Table IV presents the evaluation indices of recovered results by different methods under different SRs, including PSNR, SSIM, and running time. Similar to the results of videos, TNN maintains an advantage in terms of running time. In terms of PSNR and SSIM, LS2T2NN achieves the best performance among all competing methods. Compared with DTNN,

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

[2]The data is available at https://www.cs.columbia.edu/CAVE/databases/multispectral/

TABLE III
EVALUATION INDICES OF RECOVERY RESULTS BY DIFFERENT METHODS ON VIDEOS UNDER DIFFERENT SRS

Data	Method	SR=0.10			SR=0.15			SR=0.20			SR=0.30		
		PSNR	SSIM	Time	PSNR	SSIM	Time	PSNR	SSIM	Time	PSNR	SSIM	Time
 *Rhino*	Observed	6.53	0.06	–	6.78	0.07	–	7.04	0.08	–	7.62	0.10	–
	TNN	24.06	0.53	**16.50**	25.31	0.59	**15.41**	26.43	0.65	**14.80**	28.55	0.75	**15.16**
	RTSVD	24.31	0.53	18.36	25.57	0.60	20.80	26.62	0.65	20.34	28.69	0.75	21.08
	TQRTNN	24.21	0.54	49.75	25.52	0.60	48.76	26.61	0.65	50.53	28.67	0.75	41.86
	UTNN	24.25	0.53	150.00	25.53	0.59	147.92	26.62	0.65	141.30	28.69	0.75	110.52
	DTNN	24.50	0.57	505.79	25.80	0.63	617.73	26.96	0.68	518.70	29.08	0.77	528.51
	LS2T2NN	**26.40**	**0.63**	50.08	**28.25**	**0.71**	138.03	**29.82**	**0.78**	208.81	**31.71**	**0.84**	89.00
 *Bird*	Observed	7.74	0.02	–	7.98	0.03	–	8.25	0.04	–	8.83	0.05	–
	TNN	27.22	0.76	**16.97**	29.28	0.83	**17.17**	31.08	0.87	**17.14**	34.30	0.93	**23.05**
	RTSVD	27.54	0.77	19.91	29.65	0.83	20.97	31.38	0.88	26.02	34.51	0.93	25.74
	TQRTNN	27.41	0.77	67.59	29.55	0.83	50.87	31.33	0.88	58.06	34.46	0.93	65.71
	UTNN	27.83	0.79	158.55	30.11	0.85	151.48	31.95	0.89	159.75	35.10	0.94	130.51
	DTNN	28.55	0.83	517.04	30.60	0.88	496.70	32.51	0.91	502.04	36.17	0.95	496.61
	LS2T2NN	**30.40**	**0.85**	28.34	**33.38**	**0.91**	62.77	**35.55**	**0.94**	83.90	**38.16**	**0.97**	71.28
 *Horse*	Observed	6.53	0.01	–	6.78	0.02	–	7.04	0.02	–	7.62	0.02	–
	TNN	26.81	0.66	**17.90**	28.42	0.73	**16.87**	29.79	0.79	**16.23**	32.34	0.86	**17.23**
	RTSVD	27.09	0.67	18.71	28.71	0.74	22.50	29.96	0.79	19.32	32.46	0.87	22.58
	TQRTNN	26.93	0.67	55.91	28.64	0.74	50.72	29.95	0.79	48.49	32.45	0.87	48.15
	UTNN	27.27	0.68	170.42	29.00	0.76	152.99	30.30	0.81	119.14	32.91	0.88	115.18
	DTNN	27.52	0.74	556.79	29.42	0.80	629.70	30.98	0.85	561.31	33.61	0.91	567.97
	LS2T2NN	**29.48**	**0.76**	39.37	**31.44**	**0.83**	78.39	**33.45**	**0.88**	116.08	**36.26**	**0.93**	64.37
 *Wildcat*	Observed	3.51	0.01	–	3.75	0.01	–	4.02	0.01	–	4.60	0.01	–
	TNN	27.08	0.67	**17.13**	28.47	0.72	**16.05**	29.67	0.77	**17.66**	31.80	0.84	**15.91**
	RTSVD	27.30	0.67	18.84	28.71	0.73	20.20	29.83	0.77	20.58	31.88	0.84	22.10
	TQRTNN	27.10	0.68	50.60	28.60	0.73	49.08	29.83	0.77	52.88	31.89	0.84	46.28
	UTNN	27.47	0.69	159.58	29.00	0.75	153.69	30.23	0.79	150.33	32.33	0.86	112.55
	DTNN	28.09	0.76	631.10	29.78	0.81	696.85	31.22	0.84	538.09	33.45	0.89	559.91
	LS2T2NN	**29.84**	**0.77**	33.59	**31.55**	**0.82**	75.82	**33.06**	**0.86**	103.40	**35.21**	**0.91**	63.01
 *Flight*	Observed	4.90	0.01	–	5.15	0.01	–	5.41	0.01	–	5.99	0.02	–
	TNN	26.55	0.82	20.42	27.89	0.85	19.43	29.09	0.87	**15.09**	31.20	0.91	**15.04**
	RTSVD	26.61	0.82	**16.34**	28.10	0.85	**17.17**	29.26	0.87	17.72	31.28	0.91	17.56
	TQRTNN	26.57	0.82	46.03	28.09	0.85	51.48	29.25	0.87	48.75	31.28	0.91	50.09
	UTNN	27.01	0.84	168.12	28.66	0.87	159.16	30.00	0.89	135.67	32.52	0.93	108.21
	DTNN	27.63	0.89	579.16	29.58	0.91	586.35	31.29	0.93	540.34	34.10	0.96	622.20
	LS2T2NN	**29.99**	**0.90**	43.91	**32.04**	**0.92**	97.21	**33.70**	**0.94**	113.61	**36.67**	**0.97**	90.40

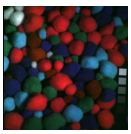


Ground truth | Observed | TNN [26] | RTSVD [43] | TQRTNN [28] | UTNN [38] | DTNN [39] | LS2T2NN

Fig. 4. The selected band of recovery results by different methods on videos under SR=0.2. From top to bottom: *Rhino*, *Bird*, *Horse*, *Wildcat*, and *Flight*, respectively.

LS2T2NN has a significant advantage in terms of running time, which is consistent with the complexity analysis. These results further validate the superiority of the proposed method in terms of computational cost and recovery performance.

TABLE IV
EVALUATION INDICES OF RESTORATION RESULTS BY DIFFERENT METHODS ON MSIS UNDER DIFFERENT SRS

Data	Method	SR=0.05			SR=0.10			SR=0.15			SR=0.20		
		PSNR	SSIM	Time	PSNR	SSIM	Time	PSNR	SSIM	Time	PSNR	SSIM	Time
 *Balloon*	Observed	13.53	0.12	–	13.76	0.15	–	14.01	0.18	–	14.27	0.20	–
	TNN	31.69	0.87	**15.77**	36.27	0.94	**11.20**	39.66	0.97	**10.84**	42.28	0.98	**16.16**
	RTSVD	32.15	0.87	36.77	36.62	0.94	19.17	39.80	0.97	15.90	42.30	0.98	43.68
	TQRTNN	31.68	0.87	28.39	36.26	0.94	20.99	39.66	0.97	33.44	42.30	0.98	27.32
	UTNN	33.78	0.92	395.53	39.28	0.97	185.32	43.20	**0.99**	122.00	46.40	**0.99**	117.31
	DTNN	35.44	0.94	415.75	40.51	0.98	405.49	44.22	**0.99**	423.61	47.42	**0.99**	425.57
	LS2T2NN	**38.35**	**0.97**	33.07	**42.99**	**0.99**	71.77	**45.58**	**0.99**	56.57	**47.97**	**0.99**	84.47
 *Beer*	Observed	9.65	0.02	–	9.89	0.03	–	10.13	0.04	–	10.40	0.05	–
	TNN	32.41	0.88	**17.28**	37.81	0.96	**13.01**	41.35	0.98	**10.33**	43.99	**0.99**	**10.45**
	RTSVD	33.19	0.89	19.33	38.17	0.96	16.88	41.52	0.98	15.91	43.98	**0.99**	16.66
	TQRTNN	32.46	0.88	41.10	37.57	0.95	29.35	41.28	0.98	29.44	43.88	**0.99**	27.08
	UTNN	35.62	0.95	220.18	41.04	**0.98**	173.65	44.66	**0.99**	121.00	47.18	**0.99**	120.63
	DTNN	35.95	0.95	416.50	41.29	0.97	402.04	45.05	**0.99**	427.96	48.47	**0.99**	454.29
	LS2T2NN	**38.67**	**0.96**	46.76	**43.36**	**0.98**	87.74	**46.25**	0.98	103.86	**48.55**	**0.99**	107.82
 *Feather*	Observed	13.35	0.18	–	13.59	0.22	–	13.83	0.26	–	14.10	0.29	–
	TNN	27.59	0.76	**20.53**	31.98	0.88	**11.44**	34.79	0.93	**9.53**	37.00	0.95	**9.17**
	RTSVD	28.03	0.76	16.75	32.04	0.88	15.76	34.83	0.93	19.17	37.00	0.95	14.54
	TQRTNN	27.57	0.76	40.72	31.94	0.88	23.06	34.78	0.93	23.78	37.07	0.95	25.43
	UTNN	28.30	0.82	158.85	33.33	0.92	162.99	36.88	0.96	108.12	39.81	**0.98**	112.85
	DTNN	29.02	0.83	389.24	33.32	0.92	408.84	36.95	0.96	419.57	39.89	0.97	431.64
	LS2T2NN	**30.99**	**0.88**	73.87	**35.39**	**0.95**	223.73	**38.45**	**0.97**	382.16	**41.34**	**0.98**	502.18
 *Pompom*	Observed	11.66	0.05	–	11.89	0.08	–	12.14	0.11	–	12.40	0.13	–
	TNN	23.34	0.56	18.07	28.39	0.76	**12.10**	31.74	0.85	**9.45**	34.39	0.91	**10.12**
	RTSVD	23.92	0.56	**16.79**	28.51	0.76	17.18	31.75	0.85	19.83	34.38	0.91	14.91
	TQRTNN	23.33	0.56	38.20	28.39	0.76	28.61	31.70	0.85	29.84	34.41	0.91	29.42
	UTNN	25.32	0.66	208.92	31.34	0.86	146.54	35.62	0.93	111.82	38.97	0.96	118.49
	DTNN	28.05	0.78	398.87	32.48	0.89	404.86	36.17	0.94	409.39	39.64	0.96	453.83
	LS2T2NN	**31.75**	**0.87**	70.88	**36.57**	**0.95**	224.03	**39.23**	**0.97**	220.73	**41.30**	**0.98**	262.09
 *Toy*	Observed	11.17	0.25	–	11.41	0.29	–	11.66	0.32	–	11.92	0.36	–
	TNN	27.38	0.82	**17.60**	31.45	0.90	**11.66**	34.42	0.94	**11.12**	36.68	0.96	**10.56**
	RTSVD	27.48	0.81	20.80	31.65	0.90	21.15	34.54	0.94	19.21	36.69	0.96	18.93
	TQRTNN	27.12	0.81	43.96	31.43	0.90	29.14	34.37	0.94	28.28	36.75	0.96	29.99
	UTNN	29.40	0.87	232.15	35.00	0.95	182.02	39.05	**0.98**	121.50	42.07	**0.99**	118.65
	DTNN	30.34	0.90	419.57	35.38	**0.96**	409.71	39.33	0.97	406.96	42.89	**0.99**	416.08
	LS2T2NN	**32.12**	**0.91**	71.56	**36.56**	**0.96**	178.21	**40.23**	**0.98**	177.90	**42.99**	**0.99**	201.94

Furthermore, we show the recovery results obtained by different methods under SR = 0.05 in Fig. 5. To better observe the restoration details of each method, we magnify a local region for each one. From Fig. 5, the proposed method recovers local image details and color fidelity better than other competing methods, which indicates that LS2T2NN has superior performance in exploring the low-rank structure of MSIs.

C. Experiments on Color Images

In the part, we choose five color images[3] of size $768 \times 512 \times 3$ to evaluate the effectiveness of the proposed method. For each image, SRs are set to be 0.10, 0.15, 0.20, and 0.30, respectively.

Table V lists the evaluation indices of recovery results by different methods on color images under different SRs. From this table, we observe that LS2T2NN has obvious advantages compared with DTNN in terms of running time. In addition, LS2T2NN significantly outperforms competing methods and yields competitive performance in terms of PSNR and SSIM. Fig. 6 shows the recovery results by TNN, RTSVD, TQRTNN, UTNN, DTNN, and LS2T2NN on color images. As observed, compared with the other methods, the images restored by LS2T2NN are the closest to the ground truth, which indicates that LS2T2NN is superior in preserving image structure and recovering local details.

[3]The data is available at http://r0k.us/graphics/kodak/

D. Discussions

1) Contribution of Spatial Projections: In this part, we show the effectiveness and superiority of learnable semi-orthogonal transforms in the proposed method. Specifically, we conduct four setting, i.e., $\boldsymbol{T}_1$ is a random semi-orthogonal matrix and $\boldsymbol{T}_2$ is an identity matrix (termed as LS2T2NN (uni-random)), both $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$ are random semi-orthogonal matrices (termed as LS2T2NN (bi-random)), $\boldsymbol{T}_1$ is a learnable semi-orthogonal matrix and $\boldsymbol{T}_2$ is an identity matrix (termed as LS2T2NN(uni-learned)), both $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$ are learnable semi-orthogonal matrices (termed as LS2T2NN (bi-learned)). Fig. 7 shows the AccEgy with the corresponding percentage of singular values of recovery results by different methods on MSIs and videos, which indicates that learned-based methods yield more concentrated singular value energies than random-based methods. Additionally, Table VI presents the evaluation metrics of restored results by different settings on MSIs under SR = 0.05 and videos under SR = 0.1. From this table, it can be observed that LS2T2NN (bi-learned) outperforms the other three methods. The above results demonstrate the following two conclusions: First, the learned-based method can obtain a lower-rank representation than the random-based method.

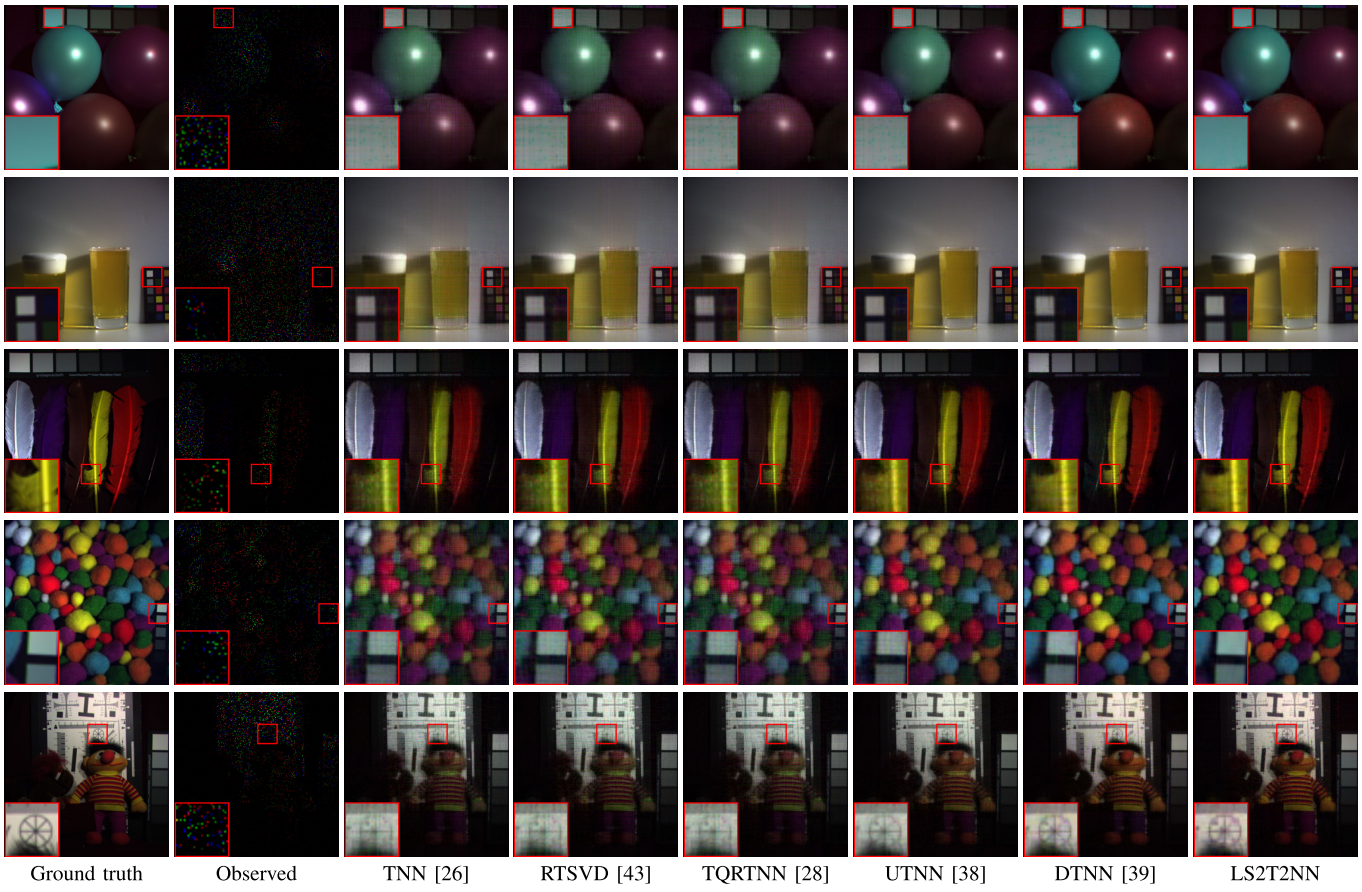


Fig. 5. The selected pseudo-color images of recovery results by different methods on MSIs under SR=0.05. From top to bottom: *Balloon*, *Beer*, *Feather*, *Pompom*, and *Toy*, respectively.

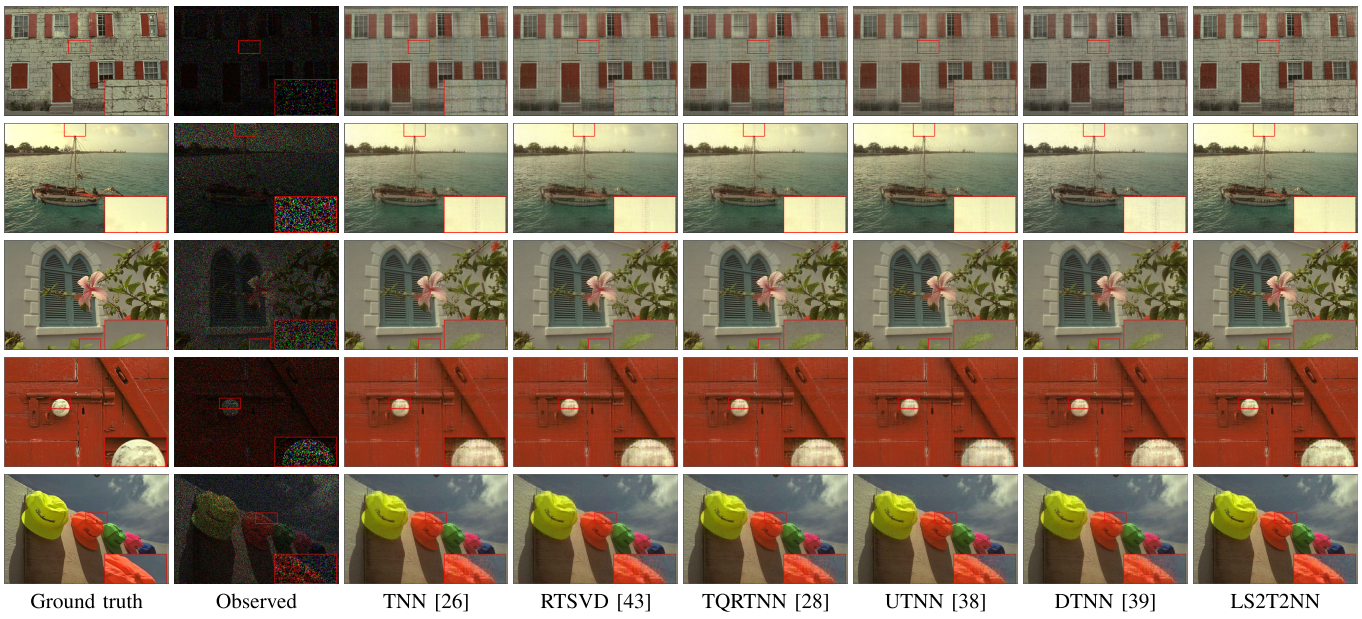


Fig. 6. The recovered color images by different methods. From left to right: the ground truth, the observed image, and the recovered images of TNN, RTSVD, TQRTNN, UTNN, DTNN, and LS2T2NN, respectively.

Second, the projection in two directions, as employed in LS2T2NN (bi-learned), is more effective than the projection in one direction used by the other methods.

2) Influence of Initializations: In this part, we investigate the influence of the initialization of $\mathcal{X}_0$ and $\boldsymbol{T}_i$ for the performance of LS2T2NN. To this end, we conduct experiments on MSI *balloon* as an example. **a)** We initialize $\mathcal{X}_0$ with the following three strategies: the observed tensor (termed as Observed), the recovered result of TNN (termed as TNN), and the linear interpolation of the observed tensor

TABLE V

EVALUATION INDICES OF RECOVERY RESULTS BY DIFFERENT METHODS ON COLOR IMAGES UNDER DIFFERENT SRS

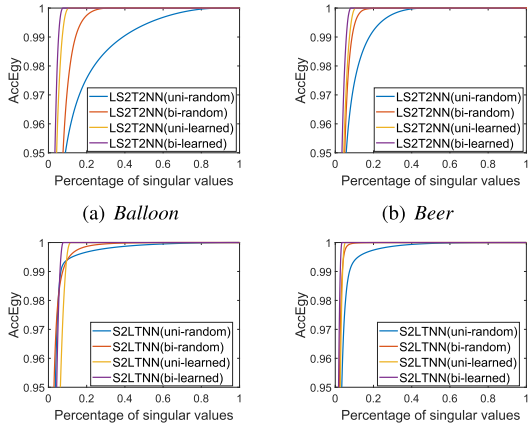
Data	Method	SR=0.10			SR=0.15			SR=0.20			SR=0.30		
		PSNR	SSIM	Time	PSNR	SSIM	Time	PSNR	SSIM	Time	PSNR	SSIM	Time
 *image01*	Observed	7.57	0.02	–	7.82	0.03	–	8.08	0.04	–	8.66	0.06	–
	TNN	21.92	0.48	**5.59**	23.31	0.59	**4.74**	24.53	0.67	**4.44**	26.96	0.79	**4.86**
	RTSVD	21.97	0.49	9.06	23.30	0.59	9.48	24.46	0.66	9.56	26.77	0.78	11.03
	TQRTNN	21.90	0.48	16.13	23.29	0.58	14.33	24.52	0.67	14.22	26.95	0.79	14.09
	UTNN	21.48	0.46	21.42	23.37	0.59	23.54	24.52	0.67	19.30	26.82	0.79	19.27
	DTNN	22.07	0.53	82.64	23.47	0.62	70.06	24.73	0.69	68.27	27.04	**0.80**	76.40
	LS2T2NN	**22.62**	**0.54**	24.85	**24.01**	**0.63**	30.25	**25.22**	**0.70**	24.49	**27.29**	**0.80**	16.84
 *image02*	Observed	5.73	0.02	–	5.98	0.02	–	6.24	0.03	–	6.82	0.04	–
	TNN	22.60	0.50	**4.79**	23.93	0.60	**4.44**	25.19	0.68	**4.16**	27.61	0.80	**4.35**
	RTSVD	22.59	0.50	8.27	23.92	0.59	8.25	25.16	0.67	8.84	27.52	0.79	9.58
	TQRTNN	22.57	0.50	15.45	23.92	0.60	13.94	25.14	0.67	13.74	27.61	0.80	14.13
	UTNN	22.65	0.51	22.99	24.03	0.60	21.84	25.26	0.68	36.28	27.66	0.80	20.97
	DTNN	22.79	0.54	72.17	24.28	0.63	70.62	25.58	0.70	70.42	27.92	**0.81**	69.06
	LS2T2NN	**23.47**	**0.56**	18.35	**24.83**	**0.65**	19.51	**26.01**	**0.71**	18.06	**28.15**	**0.81**	16.18
 *image03*	Observed	7.68	0.02	–	7.93	0.03	–	8.19	0.04	–	8.77	0.05	–
	TNN	22.12	0.51	**5.42**	24.12	0.61	**4.54**	25.75	0.69	**4.19**	28.58	0.80	**4.51**
	RTSVD	22.23	0.50	8.99	24.14	0.60	7.90	25.76	0.69	8.71	28.58	0.80	8.83
	TQRTNN	22.12	0.51	16.12	24.04	0.60	13.95	25.74	0.69	14.17	28.71	0.81	14.17
	UTNN	22.06	0.50	28.61	24.04	0.61	21.30	25.62	0.68	22.14	28.39	0.80	18.85
	DTNN	22.55	**0.61**	70.15	24.19	**0.68**	71.35	25.77	**0.74**	65.12	28.41	0.82	72.61
	LS2T2NN	**23.62**	0.55	24.92	**25.72**	0.66	19.57	**27.42**	**0.74**	19.14	**30.37**	**0.84**	16.76
 *image04*	Observed	11.80	0.04	–	12.04	0.05	–	12.31	0.07	–	12.88	0.09	–
	TNN	25.88	0.58	**6.28**	27.20	0.65	**6.16**	28.40	0.71	**6.10**	30.63	0.80	**6.08**
	RTSVD	25.97	0.57	8.84	27.24	0.64	10.93	28.50	0.70	7.95	30.68	0.80	9.33
	TQRTNN	26.00	0.60	22.51	27.44	0.66	21.09	28.54	0.71	15.70	30.74	0.80	25.40
	UTNN	25.69	0.61	31.50	26.98	0.66	29.62	28.40	0.72	16.51	30.25	0.79	17.53
	DTNN	25.35	0.63	85.55	27.17	**0.71**	85.19	28.74	0.75	89.29	31.26	**0.84**	82.70
	LS2T2NN	**26.72**	**0.64**	21.73	**28.33**	**0.71**	17.69	**29.57**	**0.76**	18.43	**31.72**	0.83	21.44
 *image05*	Observed	8.15	0.03	–	8.40	0.03	–	8.66	0.04	–	9.24	0.05	–
	TNN	23.84	0.52	**6.14**	25.59	0.60	**5.84**	27.20	0.68	**3.31**	29.94	0.80	**3.27**
	RTSVD	24.01	0.50	8.70	25.65	0.56	14.08	27.28	0.69	7.87	29.88	0.79	6.67
	TQRTNN	23.84	0.52	23.01	25.58	0.60	22.33	27.28	0.69	25.27	29.93	0.80	22.71
	UTNN	24.09	0.53	29.88	25.88	0.62	28.63	27.53	0.70	20.74	30.19	0.81	16.86
	DTNN	23.25	0.57	75.33	25.42	0.65	76.72	27.00	0.71	69.99	30.07	**0.82**	67.69
	LS2T2NN	**25.42**	**0.58**	18.99	**27.37**	**0.67**	15.15	**28.74**	**0.73**	16.51	**31.31**	**0.82**	15.86

TABLE VI

EVALUATION METRICS OF RECOVERY RESULTS BY DIFFERENT SPATIAL TRANSFORMED STRATEGIES ON MSIS AND VIDEOS

Transfrom	Data	PSNR	SSIM	Time	Data	PSNR	SSIM	Time
Observed	*Balloon*	13.52	0.12	–	*Rhino*	7.74	0.02	–
LS2T2NN(uni-random)		32.99	0.89	142.97		23.30	0.43	136.10
LS2T2NN(bi-random)		34.84	0.92	50.08		24.73	0.54	92.15
LS2T2NN(uni-learned)		37.27	0.96	138.36		25.53	0.62	103.07
LS2T2NN(bi-learned)		**38.36**	**0.97**	**37.44**		**26.40**	**0.63**	**50.10**
Observed	*Beer*	9.66	0.03	–	*Bird*	6.53	0.06	–
LS2T2NN(uni-random)		33.03	0.88	129.20		26.40	0.70	165.96
LS2T2NN(bi-random)		35.15	0.93	90.02		28.33	0.75	129.99
LS2T2NN(uni-learned)		37.59	0.95	129.61		29.67	0.84	68.49
LS2T2NN(bi-learned)		**38.66**	**0.96**	**49.94**		**30.50**	**0.85**	**30.10**

(termed as Interpolation) [59]. The evaluation metrics for different initialization strategies of $\mathcal{X}_0$ on MSI *balloon* under various SRs are reported in Table VII. From this table, we observe that the interpolation-based LS2T2NN method achieves superior results in terms of PSNR and SSIM, whereas the TNN-based LS2T2NN method takes less time under high SRs. These results imply that LS2T2NN is sensitive to the initialization of $\mathcal{X}_0$ due to the non-convex property of the proposed method. **b)** For the initialization of $\boldsymbol{T}_i$, we employed three strategies: the semi-orthogonal matrix based on the random matrix (termed as random), the unitary matrix obtained by SVD (termed as SVD), and the DCT matrix (termed as DCT). Table VIII reports that evaluation metrics of recovered results by different initializations of $\boldsymbol{T}_i$ on MSI *balloon* under different SRs. The results indicate that the performance of LS2T2NN is robust for different initializations of $\boldsymbol{T}_i$.



(a) *Balloon* (b) *Beer* (c) *Rhino* (d) *Bird*

Fig. 7. The AccEgy with the corresponding percentage of singular values of recovery results by different spatial projections.

3) Numerical Convergence Analysis: In this part, we conduct experiments on videos, MSIs, and color images to demonstrate the numerical convergence of Algorithm 1. Fig. 8 draws the relative error curves versus the number of iterations by different datasets under various SRs. As the iterations increase, the relative changes decrease towards 10^{-4} that the proposed algorithm is numerically convergent.

4) Parameters Analysis: In this part, we discuss the effects of parameters in the proposed method, including regularization parameters α and β, the proximal parameter ρ, the estimated

TABLE VII

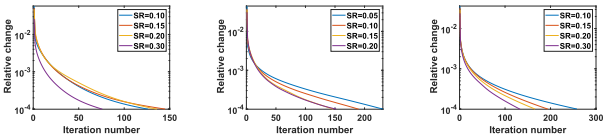
EVALUATION METRICS OF RECOVERED RESULTS BY DIFFERENT INITIALIZATIONS OF $\mathcal{X}_0$ ON MSI *balloon* UNDER DIFFERENT SRS

SR	Initialization	PSNR	SSIM	Time
0.05	Observed	32.10	0.92	74.28
	TNN	36.98	0.96	44.03
	Interpolation	**38.36**	**0.97**	**31.03**
0.10	Observed	36.19	0.96	133.27
	TNN	42.06	**0.99**	**65.53**
	Interpolation	**42.99**	**0.99**	69.41
0.15	Observed	38.94	0.98	101.59
	TNN	45.24	**0.99**	**50.27**
	Interpolation	**45.58**	**0.99**	56.37
0.20	Observed	40.71	0.98	124.00
	TNN	47.64	**0.99**	**59.83**
	Interpolation	**47.96**	**0.99**	84.33

TABLE VIII

EVALUATION METRICS OF RECOVERED RESULTS BY DIFFERENT INITIALIZATIONS OF $\boldsymbol{T}_i$ ON MSI *balloon* UNDER DIFFERENT SRS

SR	Methods	PSNR	SSIM	Time
0.05	Random	38.32	**0.97**	**31.06**
	SVD	**38.39**	**0.97**	32.90
	DCT	38.35	**0.97**	33.07
0.1	Random	**43.00**	**0.99**	**69.76**
	SVD	**43.00**	**0.99**	71.76
	DCT	**43.00**	**0.99**	71.24
0.15	Random	**45.58**	**0.99**	**55.16**
	SVD	45.57	**0.99**	56.63
	DCT	45.57	**0.99**	56.56
0.2	Random	47.94	**0.99**	**81.80**
	SVD	**47.96**	**0.99**	84.47
	DCT	**47.96**	**0.99**	84.47



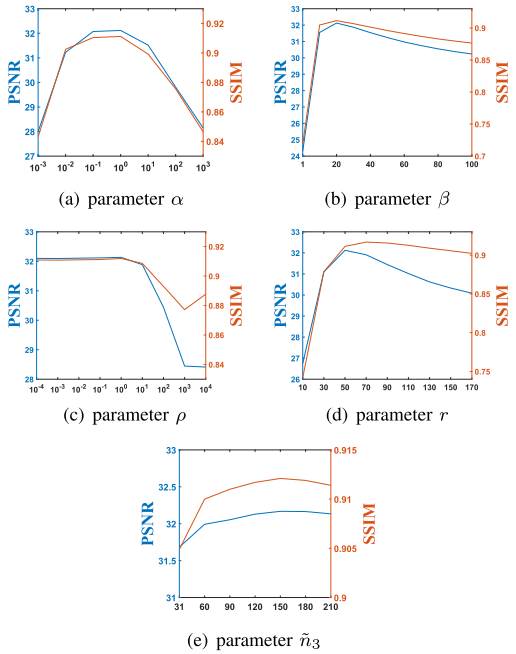
(a) video (b) MSI (c) color image

Fig. 8. The relative error curves versus the number of iterations on videos, MSIs, and color images.

rank r along spatial modes, and the parameter of learnable redundant transform $\tilde{n}_3$ along spectral mode. To evaluate the impact of these parameters, we use the MSI *toy* with SR $= 0.05$ as an example.

a) The regularization parameter α is used to balance the weight between $\|\mathcal{X} - \mathcal{E} \times_1 \boldsymbol{T}_1 \times_2 \boldsymbol{T}_2\|_F^2$ and other items. Fig. 9(a) illustrates the PSNR and SSIM values with respect to the parameter α chosen from the set $\{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}\}$. It can be observed that the optimal value of α falls within the range of $[10^{-1}, 10^{1}]$, and the best PSNR value can be obtained when α is set to 1. Therefore, for different test data and SRs, α can be selected within the range $[10^{-1}, 10^{1}]$.

b) The other regularization parameter β is utilized to balance the weight between $\|\mathcal{E} - \mathcal{Z} \times_3 \boldsymbol{D}\|_F^2$ and other items. Fig. 9(b) displays the PSNR and SSIM values with respect to the parameter β chosen from the candidate set $\{1, 10, 20, \cdots, 70, 80\}$. It can be observed that the optimal



(a) parameter α (b) parameter β

(c) parameter ρ (d) parameter r

(e) parameter $\tilde{n}_3$

Fig. 9. The PSNR and SSIM values with respect to different parameter settings on MSI *toy* under SR = 0.05.

parameter value falls within the range [10, 40], and the optimal parameter β is set to 20.

c) For the proximal parameters ρ_i, it is a challenging task to determine the optimal values of each ρ_i. Therefore, we suggest setting all proximal parameters to be equal, i.e., $\rho = \rho_1 = \rho_2 = \rho_3 = \rho_4 = \rho_5$. Fig. 9(c) shows the PSNR and SSIM values with respect to the parameter ρ chosen from the set $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}, 10^{4}\}$. We can see that a lower ρ can receive a stable result. Thus, in our experiments, ρ is set to 0.1.

d) For the spatial rank r, we empirically set $r = r_1 = r_2$ to reduce parameters. Fig. 9(d) shows the PSNR and SSIM values with respect to the parameter r selected from the candidate set $\{10, 20, \cdots, 160, 170\}$. We can see that the proposed method has the best performance when r is set to 50.

e) The parameter $\tilde{n}_3$ affects the ability of redundant transform $\boldsymbol{D}$ on capturing low-rank representation. Fig. 9(e) displays the PSNR and SSIM values with respect to the parameter $\tilde{n}_3$. We can observe from Fig. 9(e) that the best PSNR and SSIM values are attained when $\tilde{n}_3 \approx 5n_3$. Motivated by this observation, $\tilde{n}_3$ is set to $5n_3$ in our experiments.

V. CONCLUSION

This paper presents the LS2T2NN model for multi-dimensional visual data recovery, which enjoys better low-rankness characterization with the cheap computational cost. Specifically, the proposed model employs the learnable semi-orthogonal transforms to boost the spatial low-rankness while projecting the large-scale original tensor to the small-scale intrinsic tensor. On this basis, it further boosts the low-rankness of transformed frontal slices by the learnable redundant transform. In addition, we demonstrate that the learnable semi-orthogonal transforms can bring lower-rank representation from theoretical and numerical perspectives, and then discuss the relationship between LS2T2NN and

DTNN with theoretical analysis. We apply PAM to design a solving algorithm to tackle the proposed model and establish the convergence guarantee. Extensive experimental results on videos, MSIs, and color images show that the proposed method outperforms state-of-the-art competitors in terms of evaluation metrics and running time. In this work, we investigate the proposed model SL2T2NN with third-order visual data from real world, but many real visual data is higher-order, e.g., 4th-order color videos, 4th-order hyperspectral videos, and 5th-order light field images. In the future, we will consider generalizing LS2T2NN for higher-order real-world visual data.

REFERENCES

[1] T. Xu, X. Kong, Q. Shen, Y. Chen, and Y. Zhou, "Deep and low-rank quaternion priors for color image processing," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 7, pp. 3119–3132, Jul. 2023, doi: 10.1109/TCSVT.2022.3233589.

[2] H. Zeng, J. Xue, H. Q. Luong, and W. Philips, "Multimodal core tensor factorization and its applications to low-rank tensor completion," *IEEE Trans. Multimedia*, early access, Oct. 25, 2022, doi: 10.1109/TMM.2022.3216746.

[3] L. Sun et al., "Low rank component induced spatial-spectral kernel method for hyperspectral image classification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 10, pp. 3829–3842, Oct. 2020.

[4] Y. Xu, Z. Wu, J. Chanussot, and Z. Wei, "Hyperspectral computational imaging via collaborative Tucker3 tensor decomposition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 1, pp. 98–111, Jan. 2021.

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

[6] A. C. Kokaram, "On missing data treatment for degraded video and film archives: A survey and a new Bayesian approach," *IEEE Trans. Image Process.*, vol. 13, no. 3, pp. 397–415, Mar. 2004.

[7] V. Lempitsky, A. Vedaldi, and D. Ulyanov, "Deep image prior," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9446–9454.

[8] Q. Zhang, Q. Yuan, J. Li, Z. Li, H. Shen, and L. Zhang, "Thick cloud and cloud shadow removal in multitemporal imagery using progressively spatio-temporal patch group deep learning," *ISPRS J. Photogramm. Remote Sens.*, vol. 162, pp. 148–160, Apr. 2020.

[9] Y. Chen, X. Xiao, C. Peng, G. Lu, and Y. Zhou, "Low-rank tensor graph learning for multi-view subspace clustering," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 1, pp. 92–104, Jan. 2022.

[10] Y. Zhang, L. Zhang, B. Du, and S. Wang, "A nonlinear sparse representation-based binary hypothesis model for hyperspectral target detection," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 2513–2522, Jun. 2015.

[11] Y. Liu, Z. Long, H. Huang, and C. Zhu, "Low CP rank and Tucker rank tensor completion for estimating missing components in image data," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 4, pp. 944–954, Apr. 2020.

[12] Y. Zhang, Y. Wang, Z. Han, X. Chen, and Y. Tang, "Effective tensor completion via element-wise weighted low-rank tensor train with overlapping ket augmentation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 11, pp. 7286–7300, Nov. 2022.

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

[14] L. Zhuang and M. K. Ng, "FastHyMix: Fast and parameter-free hyperspectral image mixed noise removal," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 8, pp. 4702–4716, Aug. 2023, doi: 10.1109/TNNLS.2021.3112577.

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

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

[17] 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.

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

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

[20] Y. He and G. K. Atia, "Patch tracking-based streaming tensor ring completion for visual data recovery," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 12, pp. 8312–8326, Dec. 2022.

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

[22] C.-Y. Lyu, X.-L. Zhao, B.-Z. Li, H. Zhang, and T.-Z. Huang, "Multi-dimensional image recovery via fully-connected tensor network decomposition under the learnable transforms," *J. Sci. Comput.*, vol. 93, no. 2, p. 49, Nov. 2022.

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

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

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

[26] 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.

[27] Y.-B. Zheng, T.-Z. Huang, X.-L. Zhao, T.-X. Jiang, T.-H. Ma, and T.-Y. Ji, "Mixed noise removal in hyperspectral image via low-fibered-rank regularization," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 1, pp. 734–749, Jan. 2020.

[28] F. Wu, Y. Li, C. Li, and Y. Wu, "A fast tensor completion method based on tensor QR decomposition and tensor nuclear norm minimization," *IEEE Trans. Comput. Imag.*, vol. 7, pp. 1267–1277, 2021.

[29] Y.-Y. Liu, X.-L. Zhao, Y.-B. Zheng, T.-H. Ma, and H. Zhang, "Hyperspectral image restoration by tensor fibered rank constrained optimization and plug-and-play regularization," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–17, 2022.

[30] H. Qiu, Y. Wang, S. Tang, D. Meng, and Q. Yao, "Fast and provable nonconvex tensor RPCA," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 18211–18249.

[31] H. Zeng, X. Xie, and J. Ning, "Hyperspectral image denoising via global spatial-spectral total variation regularized nonconvex local low-rank tensor approximation," *Signal Process.*, vol. 178, Jan. 2021, Art. no. 107805.

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

[33] Z. Han, S. Zhang, Z. Liu, Y. Wang, J. Yao, and Y. Wang, "Tensor robust principal component analysis with side information: Models and applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 8, pp. 3713–3725, Aug. 2023, doi: 10.1109/TCSVT.2023.3239376.

[34] C. Lu, X. Peng, and Y. Wei, "Low-rank tensor completion with a new tensor nuclear norm induced by invertible linear transforms," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5989–5997.

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

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

[37] 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.

[38] G. Song, M. K. Ng, and X. Zhang, "Robust tensor completion using transformed tensor singular value decomposition," *Numer. Linear Algebra With Appl.*, vol. 27, no. 3, p. e2299, 2020.

[39] 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.

[40] B.-Z. Li, X.-L. Zhao, T.-Y. Ji, X.-J. Zhang, and T.-Z. Huang, "Nonlinear transform induced tensor nuclear norm for tensor completion," *J. Sci. Comput.*, vol. 92, no. 3, p. 83, Sep. 2022.

[41] Y. Luo, X. Zhao, D. Meng, and T. Jiang, "HLRTF: Hierarchical low-rank tensor factorization for inverse problems in multi-dimensional imaging," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 19281–19290.

[42] C. Zeng and M. K. Ng, "Decompositions of third-order tensors: HOSVD, T-SVD, and beyond," *Numer. Linear Algebra With Appl.*, vol. 27, no. 3, p. e2290, 2020.
[43] J. Zhang, A. K. Saibaba, M. E. Kilmer, and S. Aeron, "A randomized tensor singular value decomposition based on the t-product," *Numer. Linear Algebra With Appl.*, vol. 25, no. 5, p. e2179, 2018.
[44] H. Zhang, X.-L. Zhao, T.-X. Jiang, and M. K. Ng, "Constrained low-tubal-rank tensor recovery for hyperspectral images mixed noise removal by bilateral random projections," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2019, pp. 1939–1942.
[45] C. Wang, Q. Yi, X. Liao, and Y. Wang, "An improved frequent directions algorithm for low-rank approximation via block Krylov iteration," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jan. 6, 2023, doi: 10.1109/TNNLS.2022.3233243.
[46] M. Che, X. Wang, Y. Wei, and X. Zhao, "Fast randomized tensor singular value thresholding for low-rank tensor optimization," *Numer. Linear Algebra with Appl.*, vol. 29, no. 6, p. e2444, Dec. 2022.
[47] W. Dong, G. Yu, L. Qi, and X. Cai, "Practical sketching algorithms for low-rank tucker approximation of large tensors," 2023, *arXiv:2301.11598*.
[48] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, Aug. 2009.
[49] M. E. Kilmer, K. Braman, N. Hao, and R. C. Hoover, "Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging," *SIAM J. Matrix Anal. Appl.*, vol. 34, no. 1, pp. 148–172, Jan. 2013.
[50] V. de Silva and L.-H. Lim, "Tensor rank and the ill-posedness of the best low-rank approximation problem," *SIAM J. Matrix Anal. Appl.*, vol. 30, no. 3, pp. 1084–1127, Jan. 2008.
[51] J.-L. Wang, T.-Z. Huang, X.-L. Zhao, T.-X. Jiang, and M. K. Ng, "Multi-dimensional visual data completion via low-rank tensor representation under coupled transform," *IEEE Trans. Image Process.*, vol. 30, pp. 3581–3596, 2021.
[52] K. Fan and A. J. Hoffman, "Some metric inequalities in the space of matrices," *Proc. Amer. Math. Soc.*, vol. 6, no. 1, pp. 111–116, 1955.
[53] D. Geman and G. Reynolds, "Constrained restoration and the recovery of discontinuities," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 3, pp. 367–383, Mar. 1992.
[54] M. Nikolova and M. K. Ng, "Analysis of half-quadratic minimization methods for signal and image recovery," *SIAM J. Sci. Comput.*, vol. 27, no. 3, pp. 937–966, Jan. 2005.
[55] J. Bolte, S. Sabach, and M. Teboulle, "Proximal alternating linearized minimization for nonconvex and nonsmooth problems," *Math. Program.*, vol. 146, nos. 1–2, pp. 459–494, Aug. 2014.
[56] C. Bao, H. Ji, Y. Quan, and Z. Shen, "Dictionary learning for sparse coding: Algorithms and convergence analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 7, pp. 1356–1369, Jul. 2016.
[57] 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, Jan. 2010.
[58] P. H. Schönemann, "A generalized solution of the orthogonal Procrustes problem," *Psychometrika*, vol. 31, no. 1, pp. 1–10, Mar. 1966.
[59] N. Yair and T. Michaeli, "Multi-scale weighted nuclear norm image restoration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3165–3174.
[60] Q. Huynh-Thu and M. Ghanbari, "Scope of validity of PSNR in image/video quality assessment," *Electron. Lett.*, vol. 44, no. 13, pp. 800–801, Jun. 2008.
[61] 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.

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

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

Xi-Le Zhao received the M.S. and Ph.D. degrees from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2009 and 2012, respectively. From 2013 to 2014, he was a Post-Doctoral Researcher with Prof. M. Ng at Hong Kong Baptist University. From 2016 to 2017, he was a Visiting Scholar with Prof. J. Bioucas Dias at the University of Lisbon. He is currently a Professor with the School of Mathematical Sciences, UESTC. His research interests include image processing, machine learning, and scientific computing. More information can be found on his homepage https://zhaoxile.github.io/.

Haijin Zeng (Graduate Student Member, IEEE) received the M.S. degree in applied mathematics from Northwest A&F University, Yangling, China, in 2021. He is currently pursuing the Ph.D. degree with the imec, Ghent University, Ghent, Belgium. He has authored or coauthored several research articles in his research areas in top-tier journals, including IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE TRANSACTIONS ON COMPUTATIONAL IMAGING, and *Computer Vision and Image Understanding*. His research interests include image quality improvement and applications of deep learning and low-rank representation in computational imaging. He received the Third Place in ECCV 2022 Mobile Intelligent Photography and Imaging (MIPI) Challenge.

Yao Wang received the Ph.D. degree in applied mathematics from Xi'an Jiaotong University, Xi'an, China, in 2014. He is currently an Associate Professor with the School of Management, Xi'an Jiaotong University. His current research interests include statistical signal processing, high-dimensional data analysis, and machine learning.

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