Self-Supervised Nonlinear Transform-Based Tensor Nuclear Norm for Multi-Dimensional Image Recovery

Yi-Si Luo, Xi-Le Zhao, *Member, IEEE*, Tai-Xiang Jiang, *Member, IEEE*, Yi Chang, *Member, IEEE*, Michael K. Ng, *Senior Member, IEEE*, and Chao Li

***Abstract*— Recently, transform-based tensor nuclear norm (TNN) minimization methods have received increasing attention for recovering third-order tensors in multi-dimensional imaging problems. The main idea of these methods is to perform the linear transform along the third mode of third-order tensors and then minimize the nuclear norm of frontal slices of the transformed tensor. The main aim of this paper is to propose a nonlinear multilayer neural network to learn a nonlinear transform by solely using the observed tensor in a self-supervised manner. The proposed network makes use of the low-rank representation of the transformed tensor and data-fitting between the observed tensor and the reconstructed tensor to learn the nonlinear transform. Extensive experimental results on different data and different tasks including tensor completion, background subtraction, robust tensor completion, and snapshot compressive imaging demonstrate the superior performance of the proposed method over state-of-the-art methods.**

***Index Terms*— Self-supervised learning, nonlinear transform, tensor nuclear norm, multi-dimensional image.**

I. Introduction

MANY real-world images are multi-dimensional, such as hyperspectral images (HSIs), multispectral images (MSIs), and videos. However, in many applications, multi-dimensional images are incomplete or essentially degraded [6]–[9] due to irresistible factors such as low light or failure of sensors. Thus, it is of the tremendous need to recover the high-quality underlying images from the observed images, which is one of the important imaging problems [10].

Mathematically, a multi-dimensional image can be represented by a third-order tensor [11]–[22], which preserves the multi-direction structure. Since most real-world images have low-rank structures [23]–[37], the restoration of the observed image is usually formulated as the following low-rank tensor recovery problem:

$$\min_{\mathcal{X}} \lambda \, rank(\mathcal{X}) + L(\mathcal{X}, \mathcal{O}), \tag{1}$$

where $\mathcal{O}$ denotes the observed tensor, $\mathcal{X}$ denotes the underlying low-rank tensor, $L(\mathcal{X}, \mathcal{O})$ is the fidelity loss function, and λ is the trade-off parameter.

Different from matrices, the definition of tensor rank is not unique [11], [38]. Several definitions of tensor ranks are proposed. The CP rank (see for example [11]) is defined as the smallest number of rank one tensor decomposition. However, computing the CP rank is an NP-hard problem and its convex surrogate is not clear. The Tucker rank was studied for tensors by considering the ranks of unfolding matrices from tensors, see for example [11]. However, the sum of the nuclear norm of unfolding matrices is not the convex envelope of the sum of the rank of unfolding matrices [39]. In this paper, we focus on the tensor tubal-rank [40]. The tensor tubal-rank is based on the tensor singular value decomposition (t-SVD) [41], which has been applied to various applications such as clustering [30], feature extraction [42], and super-resolution [38], [43]. The minimization of the tubal-rank is an NP-hard problem. Zhang *et al.* [1] built a convex surrogate of the tensor tubal-rank, named the tensor nuclear norm (TNN). Thus, model (1) is re-formulated as follows:

$$\min_{\mathcal{X}} \lambda \, \|\mathcal{X}\|_{\mathrm{TNN}} + L(\mathcal{X}, \mathcal{O}). \tag{2}$$

Note that the TNN of a tensor is computed by summing the nuclear norm of each transformed frontal slice where a transform is applied along the third mode of the tensor [1]. Thus, model (2) can be re-formulated as follows:

$$\min_{\mathcal{X}} \lambda \sum_k \left\| l(\mathcal{X})^{(k)} \right\|_* + L(\mathcal{X}, \mathcal{O}), \tag{3}$$

Manuscript received June 17, 2021; revised December 14, 2021 and February 24, 2022; accepted May 12, 2022. Date of publication May 24, 2022; date of current version June 2, 2022. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61876203, Grant 12171072, Grant 12001446, and Grant 62101294; in part by the Applied Basic Research Project of Sichuan Province under Grant 2021YJ0107; in part by the Key Project of Applied Basic Research in Sichuan Province under Grant 2020YJ0216; in part by the National Key Research and Development Program of China under Grant 2020YFA0714001; in part by the Fundamental Research Funds for the Central Universities under Grant JBK2202049 and Grant JBK2102001; in part by the Hong Kong Research Grant Council (HKRGC) through the General Research Fund (GRF) under Grant 12300218, Grant 12300519, Grant 17201020, Grant 17300021, Grant C1013-21GF, and Grant C7004-21GF; and in part by the Joint NSFC-RGC under Grant N-HKU76921. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Chaker Larabi.
(Corresponding author: Xi-Le Zhao.)

Yi-Si Luo and Xi-Le Zhao are with the School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: yisiluo1221@foxmail.com; xlzhao122003@163.com).

Tai-Xiang Jiang is with the School of Economic Information Engineering, Southwestern University of Finance and Economics, Chengdu 610074, China (e-mail: taixiangjiang@gmail.com).

Yi Chang is with the School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: yichang@hust.edu.cn).

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

Chao Li is with RIKEN Center on Advanced Intelligence Project, Tokyo 103-0027, Japan (e-mail: chao.li@riken.jp).
Digital Object Identifier 10.1109/TIP.2022.3176220

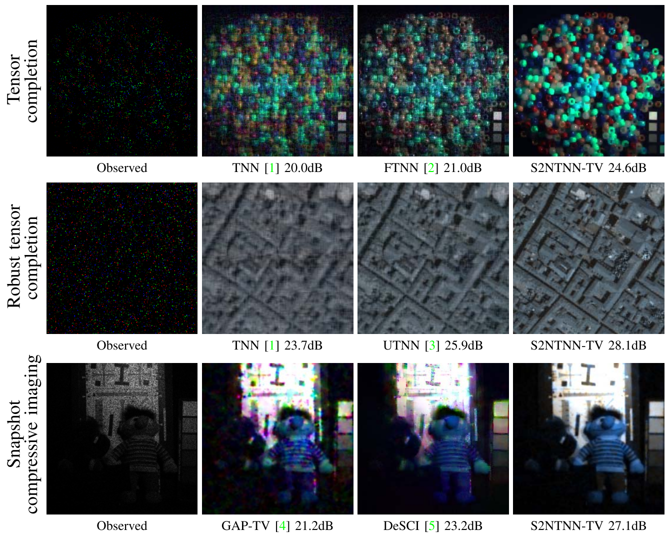


Fig. 1. The recovered results and PSNR values by different methods. Three rows respectively list the recovered results for tensor completion on MSI *Beads* with SR $= 0.05$, the recovered results for robust tensor completion on HSI *Pavia* with SR $= 0.05$, and the recovered results for snapshot compressive imaging on MSI *Toys* with SR $= 0.25$. The proposed S2NTNN-TV obtains the best PSNR values and qualitative results compared with state-of-the-art methods.

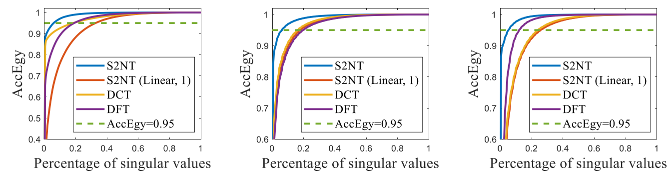


Fig. 2. The accumulation energy ratio (AccEgy) [31] with respect to the percentage of singular values of the transformed frontal slices of HSI *Pavia*, HSI *WDC mall*, and MSI *Beads*. The AccEgy is defined as $\sum_{i=1}^{k} \sigma_i^2 / \sum_j \sigma_j^2$, where σ_i is the i-th singular value. We can observe that S2NT obtains a better low-rank representation whose energy is concentrated in the larger singular values. Thus, the corresponding S2NTNN could achieve more promising results.

where $l(\mathcal{X})$ is the transformed tensor under the transform l and the superscript refers to the k-th frontal slice of the transformed tensor. More precisely, the discrete Fourier transform (DFT) is used, see [1], [41], [44]. Since TNN is convex, model (2) can be addressed by many optimization algorithms such as the alternating direction method of multipliers (ADMM).

In the literature, other transforms were considered and studied, for instance, the use of discrete cosine transform (DCT) [45], [46] for real arithmetic computation and other unitary transforms [3]. The motivation is that when a suitable transform is applied to the third-mode of a tensor, a better low-rank representation of the transformed tensor can be obtained, and therefore the underlying low-rank tensor can be more easily recovered, see [3], [47].

To explore a better low-rank representation of the transformed tensor, Jiang *et al.* [2] suggested to use the non-invertible framelet transform (a redundant basis) to represent low-rank transformed tensors. Along this research direction, data-adaptive transforms were proposed and studied. Kong *et al.* [48] proposed the data-dependent transform to capture the low Q-rank tensor structure. Jiang *et al.* [49] proposed to learn low-rank coding coefficients using dictionary approach. Ng *et al.* [50] used the left singular vectors of the unfolding matrix to establish the patched-tube unitary transform.

Nevertheless, all of the aforementioned transforms are linear which may limit their capability to model the nonlinear nature of real-world data. In this paper, we embed a nonlinear transform into the TNN. The proposed nonlinear transform consists of multiple linear transforms and nonlinear activation functions. Thus, this nonlinear transform can be interpreted as a nonlinear multilayer neural network. By optimizing the nuclear norm of the transformed frontal slices and the data-fitting between the observed tensor and the reconstructed tensor, the nonlinear transform can be learned by solely using the observed data in a self-supervised manner. We call such transform to be the Self-Supervised Nonlinear Transform (S2NT). Based on the universal approximation theorem of neural networks [51], the proposed S2NT could approximate to any functions and thus it can obtain a better and lower-rank transformed tensor (see Fig. 2), which is crucial to obtain a better recovery performance [2].

Based on the S2NT, we propose the S2NT-based TNN (S2NTNN) model for low-rank tensor recovery. The proposed S2NTNN model only includes the observed data without additional training data. Thus, the parameters of the S2NT are learned in a self-supervised manner, and the underlying low-rank tensor can be readily obtained.

Generally, only considering the low-rankness of tensor data is not sufficient to recover the multi-dimensional images with complex image details. Thus, we combine the proposed S2NTNN with the simple and efficient total variation (TV) regularization, and form the S2NTNN-TV model for low-rank tensor recovery. The TV regularization can explore the spatial local smoothness of the tensor, which improves the recovery quality. In Fig. 3, we describe a tensor completion process by using the proposed S2NTNN-TV.

We summarize the contributions of this paper as follows:

- To exploit the nonlinear nature of multi-dimensional images, we propose the S2NT-based TNN for multi-dimensional image recovery. As compared with linear transforms, the nonlinear modeling capability of S2NT is believed to obtain a better low-rank representation under the TNN framework, which is beneficial for a better recovery performance. By solely using the observed data, the parameters of the nonlinear transforms are self-supervisedly learned and the recovered result can be readily obtained.
- The proposed method is comprehensively evaluated on different data (HSIs, MSIs, and videos) and different tasks (tensor completion, background subtraction, robust tensor completion (RTC), and snapshot compressive imaging (SCI)), which validates its generalization ability and wide applicability. The superiority of our method is demonstrated as compared with state-of-the-art methods including linear transform-based TNN methods.

The outline of this paper is given as follows. In Sec. II, we introduce some related work. In Sec. III, we give preliminaries of tensors. In Sec. IV, we present the proposed method. In Sec. V, we report experimental results on different tasks. Finally, some concluding remarks are given in Sec. VI.

II. Related Work

In the literature, there were other matrix/tensor recovery methods that utilized deep or nonlinear transforms. Li *et al.* [27] introduced multiple linear transforms in the low-rank matrix completion model. Arora *et al.* [52] studied the deep linear matrix factorization and its implicit regularization for matrix completion. Fan and Chow [53] used a nonlinear function to transform the data into a feature space and then considered the nuclear norm minimization on the feature space for matrix completion. Fan and Cheng [54] suggested the deep nonlinear matrix factorization via a deep neural network for matrix completion. Based on the work proposed in [52], Li *et al.* [55] introduced the TV regularization in the deep matrix factorization for matrix completion.

Recently, nonlinear tensor recovery methods were proposed. Ma *et al.* [56] proposed the deep tensor ADMM-Net for SCI, which cleverly unfolded a TNN optimization algorithm into a nonlinear neural network. This deep tensor ADMM-Net learned a linear transform under the TNN framework. Chen and Li [57] proposed the nonlinear CP factorization and nonlinear Tucker factorization for the recommender system. Zhang *et al.* [58] learned the tensor low-rank prior to promote the reconstruction quality of SCI. All the mentioned tensor recovery methods need supervised learning and pairs of training data. In our work, we consider the classical TNN framework, which was firstly suggested by Kilmer *et al.* [41]. The transform is a key module in the TNN to exploit the interactions of frontal slices. We employ the nonlinear transform to help obtain a better low-rank representation, which can boost the recovery performance. Meanwhile, our nonlinear transform is self-supervisedly learned by solely using the observed data, which benefits its wide applicability for different tasks.

III. Preliminaries

The primary notations used in this paper are introduced in Table I. In addition, we introduce the following definitions and theorems.

Definition 1 (t-product [41]): The tensor-tensor product $\mathcal{C}=\mathcal{A}*\mathcal{B}$ is defined by $\mathcal{C}(i,j,:)=\sum_{k=1}^{n_2}\mathcal{A}(i,k,:)*\mathcal{B}(k,j,:)$, where $*$ denotes the circular convolution between two vectors.

Definition 2 (Conjugate Transpose [41]): The conjugate transpose of $\mathcal{A}\in\mathbb{R}^{n_1\times n_2\times n_3}$, denoted as $\mathcal{A}^H$, is defined by $(\mathcal{A}^H)^{(1)}=(\mathcal{A}^{(1)})^H$ and $(\mathcal{A}^H)^{(i)}=(\mathcal{A}^{(n_3+2-i)})^H(i=2,\cdots,n_3)$.

Definition 3 (Identity Tensor [41]): $\mathcal{I}\in\mathbb{R}^{n_1\times n_1\times n_3}$ is called an identity tensor if $\mathcal{I}^{(1)}$ is an identity matrix and $\mathcal{I}^{(k)}(k=2,\cdots,n_3)$ are zero matrices.

Definition 4 (Orthogonal Tensor [41]): The tensor $\mathcal{Q}$ is orthogonal if $\mathcal{Q}*\mathcal{Q}^H=\mathcal{Q}^H*\mathcal{Q}=\mathcal{I}$. $\mathcal{A}\in\mathbb{R}^{n_1\times n_2\times n_3}$ is f-diagonal if $\mathcal{A}^{(i)}(i=1,\cdots,n_3)$ are diagonal matrices.

TABLE I
Notations Used in This Paper

Notations	Interpretations
$\mathbf{X},\mathcal{X}$	matrix, tensor
$\mathcal{X}_{ijk}$	the i,j,k-th element of $\mathcal{X}$
$\mathcal{X}(:,:,k)$ or $\mathcal{X}^{(k)}$	the k-th frontal slice of $\mathcal{X}$
$\mathcal{X}(i,j,:)$	the i,j-th tube of $\mathcal{X}$
∇_d	the difference operator along the d-th dimension ($d=1,2$)
$\Vert\mathbf{X}\Vert_*$	the nuclear norm of $\mathbf{X}$
$\Vert\mathcal{X}\Vert_F$	the tensor Frobenius norm $\Vert\mathcal{X}\Vert_F=\sqrt{\langle\mathcal{X},\mathcal{X}\rangle}=\sqrt{\sum_{ijk}\mathcal{X}_{ijk}^2}$
$\Vert\mathcal{X}\Vert_{\ell_1}$	the tensor ℓ_1-norm $\Vert\mathcal{X}\Vert_{\ell_1}=\sum_{ijk}\vert\mathcal{X}_{ijk}\vert$
$\texttt{unfold}_3(\cdot)$	the mode-3 unfolding operator $\texttt{unfold}_3(\cdot):\mathbb{R}^{n_1\times n_2\times n_3}\to\mathbb{R}^{n_3\times n_1n_2}$
$\texttt{fold}_3(\cdot)$	the mode-3 folding operator $\texttt{fold}_3(\cdot):\mathbb{R}^{n_3\times n_1n_2}\to\mathbb{R}^{n_1\times n_2\times n_3}$
$\times_3$	the mode-3 tensor-matrix product $\mathcal{X}\times_3\mathbf{A}=\texttt{fold}_3(\mathbf{A}\texttt{unfold}_3(\mathcal{X}))$

Theorem 1 (t-SVD [41]): Any $\mathcal{A}\in\mathbb{R}^{n_1\times n_2\times n_3}$ can be decomposed as $\mathcal{A}=\mathcal{U}*\mathcal{S}*\mathcal{V}^H$, where $\mathcal{U}\in\mathbb{R}^{n_1\times n_1\times n_3}$ and $\mathcal{V}\in\mathbb{R}^{n_2\times n_2\times n_3}$ are orthogonal and $\mathcal{S}\in\mathbb{R}^{n_1\times n_2\times n_3}$ is f-diagonal.

Definition 5 (Tensor Tubal-Rank [41]): Given the t-SVD: $\mathcal{A}=\mathcal{U}*\mathcal{S}*\mathcal{V}^H$, where $\mathcal{A}\in\mathbb{R}^{n_1\times n_2\times n_3}$, the tubal-rank $rank_t(\mathcal{A})$ is defined as the number of nonzero singular tubes of $\mathcal{S}$.

Definition 6 (TNN [41]): The tensor nuclear norm of $\mathcal{A}\in\mathbb{R}^{n_1\times n_2\times n_3}$ is defined as $\Vert\mathcal{A}\Vert_{\text{TNN}}=\sum_{k=1}^{n_3}\left\Vert(\mathcal{A}\times_3\mathbf{F})^{(k)}\right\Vert_*$, where $\mathbf{F}\in\mathbb{R}^{n_3\times n_3}$ denotes the DFT matrix.

IV. The Proposed Method

In this section, we introduce the structure of the proposed S2NT. Using the S2NT, we build the S2NTNN model and the corresponding algorithm for low-rank tensor recovery. We employ the TV regularization and form the S2NTNN-TV model. To tackle the S2NTNN-TV model, we apply the ADMM algorithm.

A. The Structure of S2NT

Classical linear transforms in the t-SVD framework are generally represented by matrices, e.g., the DFT matrix [1], the DCT matrix [45], or the data-dependent matrix [48].

Under the motivation of building a nonlinear transform in the TNN, we propose to use a multilayer nonlinear transform. Specifically, we suggest the nonlinear mode-3 fully connected (NoFC_3) layer as the unit of S2NT. A single NoFC_3 layer is formulated as

$$w_i(\mathcal{X})=\sigma(\mathcal{X}\times_3\mathbf{W}_i),\tag{4}$$

where $\sigma(\cdot)$ denotes the nonlinear activation function and $\mathbf{W}_i$ is a learnable matrix. In this paper, we use the LeakyReLU [59] as the nonlinear activation function $\sigma(\cdot)$. Consistent with

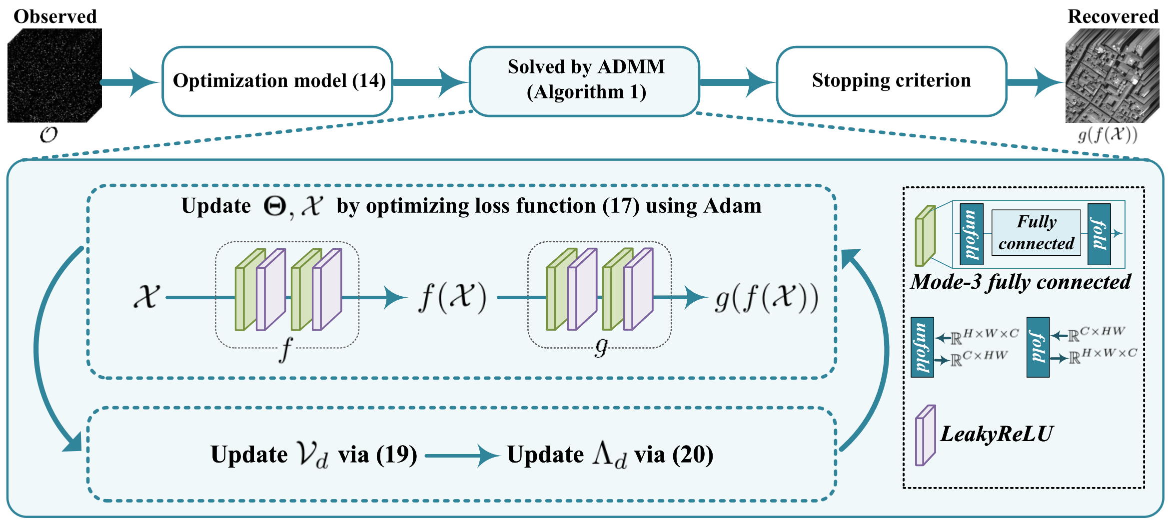


Fig. 3. The pipeline of the proposed S2NTNN-TV for multi-dimensional image recovery. Compared with traditional linear transform-based TNN family methods, our method employs nonlinear transforms f and g with high representation abilities to help obtain a better low-rank representation, leading to a promising improvement of the recovery performance.

the classical TNN, we employ the transform along mode-3 to explore the interaction of frontal slices.

We stack p NoFC$_3$ layers to build the proposed S2NT $f: \mathbb{R}^{n_1\times n_2\times n_3} \to \mathbb{R}^{n_1\times n_2\times \tilde{n}_3}$, which is formulated as

$$f(\cdot) \triangleq w_p \circ w_{p-1} \circ \cdots \circ w_1(\cdot), \tag{5}$$

where $\circ$ denotes the composition of functions and p denotes the number of NoFC$_3$ layers in f. Here, a larger $\tilde{n}_3$ can bring redundancy of the transform to obtain a better low-rank representation [2]. Similarly, we stack q NoFC$_3$ layers and develop the inverse transform $g: \mathbb{R}^{n_1\times n_2\times \tilde{n}_3} \to \mathbb{R}^{n_1\times n_2\times n_3}$, which is formulated as

$$g(\cdot) \triangleq w_{p+q} \circ w_{p+q-1} \circ \cdots \circ w_{p+1}(\cdot). \tag{6}$$

Here, the learnable parameters of f and g are the matrices $\{\mathbf{W}_i\}_{i=1}^{p+q}$. For simplicity, we use $\mathbf{\Theta} \triangleq \{\mathbf{W}_i\}_{i=1}^{p+q}$ to denote the learnable parameters of f and g.

B. S2NTNN for Low-Rank Tensor Recovery

1) Optimization Model: Next, we use the proposed S2NT f and the inverse transform g to form the S2NTNN model for low-rank tensor recovery. Given the observed data $\mathcal{O} \in \mathbb{R}^{n_1\times n_2\times n_3}$, the proposed model is formulated as

$$\min_{\mathbf{\Theta},\mathcal{X}} \lambda \sum_{k=1}^{\tilde{n}_3} \left\| (f(\mathcal{X}))^{(k)} \right\|_* + L(g(f(\mathcal{X})), \mathcal{O}), \tag{7}$$

where $f: \mathbb{R}^{n_1\times n_2\times n_3} \to \mathbb{R}^{n_1\times n_2\times \tilde{n}_3}$ and $g: \mathbb{R}^{n_1\times n_2\times \tilde{n}_3} \to \mathbb{R}^{n_1\times n_2\times n_3}$ are the nonlinear transforms defined by Eq. (5) and Eq. (6). $\mathbf{\Theta} \triangleq \{\mathbf{W}_i\}_{i=1}^{p+q}$ are the learnable parameters of f and g. $\mathcal{X} \in \mathbb{R}^{n_1\times n_2\times n_3}$ is the estimated tensor, which is also self-supervisedly optimized with the transform parameters. The recovered result is finally obtained by $g(f(\mathcal{X}))$, which is consistent to the observed $\mathcal{O}$.

In model (7), $\sum_{k=1}^{\tilde{n}_3} \left\| (f(\mathcal{X}))^{(k)} \right\|_*$ is the S2NTNN regularization and $L(g(f(\mathcal{X})), \mathcal{O})$ is the fidelity term. The fidelity term has different formulations for different recovery problems. Our model only utilizes the observed data $\mathcal{O}$ without additional training data. Thus, the parameters of the nonlinear transforms f and g are inferred in a self-supervised manner. We remark here that, given pairs of training data, we can consider the end-to-end loss form as in supervised learning framework [56]. However, since pairs of datasets are not always available for multi-dimensional images including videos and MSIs, we believe that the self-supervised method without pairs of training data is more applicable in this scenario.

2) Algorithm: Let $\mathcal{L}_1 = \lambda \sum_k \left\| (f(\mathcal{X}))^{(k)} \right\|_*$ and $\mathcal{L}_2 = L(g(f(\mathcal{X})), \mathcal{O})$, the loss function corresponding to (7) is

$$\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2. \tag{8}$$

It is expected to minimize $\mathcal{L}$ via updating the f and g parameters $\mathbf{\Theta} \triangleq \{\mathbf{W}_i\}_{i=1}^{p+q}$ and the estimated tensor $\mathcal{X}$. Due to the non-convexity of (7), we directly use the gradient descent algorithm to update $\mathbf{\Theta}$ and $\mathcal{X}$. The gradient of $\mathcal{L}_1$ on the u, v-th element of $\mathbf{W}_i$ is

$$\begin{aligned} \frac{\partial \mathcal{L}_1}{\partial (\mathbf{W}_i)_{uv}} &= \lambda \sum_k \frac{\partial \left\| (f(\mathcal{X}))^{(k)} \right\|_*}{\partial (\mathbf{W}_i)_{uv}} \\ &= \lambda \sum_k \sum_{s,t} \frac{\partial \left\| (f(\mathcal{X}))^{(k)} \right\|_*}{\partial ((f(\mathcal{X}))^{(k)})_{st}} \frac{\partial ((f(\mathcal{X}))^{(k)})_{st}}{\partial (\mathbf{W}_i)_{uv}}. \end{aligned} \tag{9}$$

The subgradient of the nuclear norm [60] is

$$\widetilde{\mathbf{U}}_k \widetilde{\mathbf{V}}_k^T \in \frac{\partial \left\| (f(\mathcal{X}))^{(k)} \right\|_*}{\partial (f(\mathcal{X}))^{(k)}}, \tag{10}$$

where $(f(\mathcal{X}))^{(k)} = \mathbf{U}_k \mathbf{S}_k \mathbf{V}_k^T$ is the matrix singular value decomposition and $\widetilde{\mathbf{U}}_k$, $\widetilde{\mathbf{V}}_k$ are $\mathbf{U}_k$, $\mathbf{V}_k$ truncated to the first s_k columns and rows. Here, s_k denotes the number of non-zero elements in $\mathbf{S}_k$. Integrating (9) and (10), we have

$$\lambda \sum_k \sum_{s,t} (\widetilde{\mathbf{U}}_k \widetilde{\mathbf{V}}_k^T)_{st} \frac{\partial ((f(\mathcal{X}))^{(k)})_{st}}{\partial (\mathbf{W}_i)_{uv}} \in \frac{\partial \mathcal{L}_1}{\partial (\mathbf{W}_i)_{uv}}. \tag{11}$$

Similarly, the gradient of $\mathcal{L}_1$ on the u, v, w-th element of $\mathcal{X}$ is

$$\lambda \sum_k \sum_{s,t} (\widetilde{\mathbf{U}}_k \widetilde{\mathbf{V}}_k^T)_{st} \frac{\partial((f(\mathcal{X}))^{(k)})_{st}}{\partial(\mathcal{X})_{uvw}} \in \frac{\partial \mathcal{L}_1}{\partial(\mathcal{X})_{uvw}}. \quad (12)$$

The gradients of $\mathcal{L}_2$ on $\mathbf{W}_i$ and $\mathcal{X}$ are

$$\begin{cases} \frac{\partial \mathcal{L}_2}{\partial(\mathbf{W}_i)_{uv}} = \sum_{r,s,t} \frac{\partial \mathcal{L}_2}{\partial(g(f(\mathcal{X})))_{rst}} \frac{\partial(g(f(\mathcal{X})))_{rst}}{\partial(\mathbf{W}_i)_{uv}} \\ \frac{\partial \mathcal{L}_2}{\partial(\mathcal{X})_{uvw}} = \sum_{r,s,t} \frac{\partial \mathcal{L}_2}{\partial(g(f(\mathcal{X})))_{rst}} \frac{\partial(g(f(\mathcal{X})))_{rst}}{\partial(\mathcal{X})_{uvw}}. \end{cases} \quad (13)$$

With these gradients, the S2NTNN model (7) can be addressed by most gradient descent-based algorithms. In this paper, we adopt the adaptive moment estimation (Adam) [61]. We set a maximum iteration number t_{max} as the stopping criterion of the Adam optimization.

Since model (7) is non-convex, the initialization of $\mathbf{\Theta}$ and $\mathcal{X}$ is important. We use the default normal distribution in PyTorch[1] to initialize the transform parameters $\mathbf{\Theta}$. We employ an initialization function $\text{Init}(\cdot)$ to initialize $\mathcal{X} = \text{Init}(\mathcal{O})$. The function $\text{Init}(\cdot)$ is chosen based on different recovery problems, specified in the experiments.

C. S2NTNN-TV for Tensor Recovery

In model (7), we only consider low-rankness of tensor data, which would be sometimes not sufficient to explore the spatial local similarity of data. Thus, we propose the TV regularized S2NTNN model for tensor recovery. The TV can explore the spatial local smoothness to improve the multi-dimensional image recovery performance.

1) Optimization Model: By introducing TV regularization in model (7), the proposed S2NTNN-TV model for tensor recovery is

$$\min_{\mathbf{\Theta},\mathcal{X}} \tau \sum_{d=1,2} \left\| \nabla_d\big(g(f(\mathcal{X}))\big) \right\|_{\ell_1} + \lambda \sum_{k=1}^{\tilde{n}_3} \left\| (f(\mathcal{X}))^{(k)} \right\|_* + L(g(f(\mathcal{X})), \mathcal{O}), \quad (14)$$

where $\sum_{d=1,2} \left\| \nabla_d\big(g(f(\mathcal{X}))\big) \right\|_{\ell_1}$ is the spatial TV regularization and τ is the weight parameter of the TV regularization. The recovered result is obtained through $g(f(\mathcal{X}))$.

2) Algorithm: To address the model (14), we apply the efficient ADMM algorithm [62]. By introducing auxiliary variables $\mathcal{V}_d$ $(d = 1, 2)$, we re-formulate model (14) as

$$\begin{aligned} &\min_{\mathbf{\Theta},\mathcal{X},\mathcal{V}_d} \tau \sum_d \|\mathcal{V}_d\|_{\ell_1} + \lambda \sum_{k=1}^{\tilde{n}_3} \left\| (f(\mathcal{X}))^{(k)} \right\|_* + L(g(f(\mathcal{X})), \mathcal{O}) \\ &\text{s.t. } \mathcal{V}_d = \nabla_d\big(g(f(\mathcal{X}))\big), \quad d = 1, 2. \end{aligned} \quad (15)$$

[1]https://pytorch.org/docs/stable/nn.init.html

Algorithm 1 S2NTNN-TV for Tensor Recovery

Input: The observed tensor $\mathcal{O}$; trade-off parameters τ and λ; Lagrange parameter β; maximum iteration t_{max}.
Initialization: $\mathcal{X} = \text{Init}(\mathcal{O})$, $\mathcal{V}_d = \nabla_d \mathcal{X}$, $\Lambda_d = \mathbf{0}$, $t = 0$.
while $t < t_{max}$ **do**
  Update $\{\mathbf{\Theta}, \mathcal{X}\}$ via (17) using Adam;
  Update $\mathcal{V}_d$ via Eq. (19);
  Update Λ_d via Eq. (20);
  t=t+1;
end while
Output: The recovered tensor $g(f(\mathcal{X}))$.

The augmented Lagrangian function of (15) is

$$\begin{aligned} L_\beta(\mathbf{\Theta}, \mathcal{X}, \mathcal{V}_d, \Lambda_d) = &\tau \sum_d \|\mathcal{V}_d\|_{\ell_1} + \lambda \sum_{k=1}^{\tilde{n}_3} \left\| (f(\mathcal{X}))^{(k)} \right\|_* \\ &+ L(g(f(\mathcal{X})), \mathcal{O}) + \sum_d \Big(\langle \Lambda_d, \nabla_d\big(g(f(\mathcal{X}))\big) - \mathcal{V}_d \rangle \\ &+ \frac{\beta}{2} \left\| \nabla_d\big(g(f(\mathcal{X}))\big) - \mathcal{V}_d \right\|_F^2 \Big), \end{aligned} \quad (16)$$

where β is the penalty parameter and Λ_d is the Lagrangian multiplier. Under the framework of ADMM, the joint minimization problem can be decomposed into easier sub-problems, followed by the update of Lagrangian multipliers.

a) $\{\mathbf{\Theta}, \mathcal{X}\}$ sub-problem: The $\{\mathbf{\Theta}, \mathcal{X}\}$ sub-problem is

$$\begin{aligned} \min_{\mathbf{\Theta},\mathcal{X}} \lambda \sum_{k=1}^{\tilde{n}_3} \left\| (f(\mathcal{X}))^{(k)} \right\|_* &+ L(g(f(\mathcal{X})), \mathcal{O}) \\ &+ \frac{\beta}{2} \sum_d \left\| \nabla_d\big(g(f(\mathcal{X}))\big) - \mathcal{V}_d + \frac{\Lambda_d}{\beta} \right\|_F^2. \end{aligned} \quad (17)$$

Similar to the optimization of (7), we update $\mathbf{\Theta}$ and $\mathcal{X}$ by the Adam algorithm. Since the estimated intermediate variables $\mathcal{V}_d$ and Λ_d may not be accurate enough, it is not necessary to use an exact solution of (17). Thus, we employ one step of the Adam algorithm to update $\mathbf{\Theta}, \mathcal{X}$ at each iteration of the ADMM algorithm [62]–[64] for computational efficiency.

b) $\mathcal{V}_d$ sub-problem: The $\mathcal{V}_d$ sub-problem $(d = 1, 2)$ is

$$\min_{\mathcal{V}_d} \tau \|\mathcal{V}_d\|_{\ell_1} + \frac{\beta}{2} \left\| \mathcal{V}_d - \left(\nabla_d(g(f(\mathcal{X}))) + \frac{\Lambda_d}{\beta} \right) \right\|_F^2, \quad (18)$$

which can be exactly solved by

$$\mathcal{V}_d = \texttt{Soft}_{\frac{\tau}{\beta}} \left(\nabla_d(g(f(\mathcal{X}))) + \frac{\Lambda_d}{\beta} \right), \quad (19)$$

where $\texttt{Soft}_v(\cdot)$ denotes the soft-thresholding operator with threshold value v.

c) Λ_d updating: The multipliers Λ_d $(d = 1, 2)$ are updated by

$$\Lambda_d = \Lambda_d + \beta \Big(\nabla_d(g(f(\mathcal{X}))) - \mathcal{V}_d \Big). \quad (20)$$

Moreover, we set a maximum iteration number t_{max} as the stopping criterion of the ADMM algorithm. The ADMM algorithm for solving model (14) is summarized in Algorithm 1.

TABLE II

THE QUANTITATIVE RESULTS BY DIFFERENT METHODS ON DIFFERENT DATA FOR TENSOR COMPLETION. THE **BEST** VALUE ARE HIGHLIGHTED BY **BOLDFACE**, AND THE SECOND-BEST VALUE ARE HIGHLIGHTED BY UNDERLINED

Data	SR	0.05			0.1			0.15			0.2			0.25		
	Metric	PSNR	SSIM	SAM	PSNR	SSIM	SAM	PSNR	SSIM	SAM	PSNR	SSIM	SAM	PSNR	SSIM	SAM
HSI *WDC mall* (256×256×191)	Observed	14.567	0.076	1.351	14.801	0.118	1.253	15.050	0.158	1.176	15.312	0.199	1.109	15.594	0.239	1.049
	TRLRF [65]	27.044	0.854	0.209	29.463	0.912	0.164	29.959	0.920	0.160	29.671	0.918	0.168	30.589	0.931	0.156
	TNN [1]	29.513	0.916	0.197	33.249	0.962	0.144	36.109	0.979	0.113	38.311	0.986	0.093	40.075	0.990	0.079
	FTNN [2]	32.776	0.955	0.131	37.752	0.983	0.095	41.311	0.991	0.074	43.874	0.994	0.062	45.954	0.996	0.053
	S2NTNN	40.118	0.992	0.055	44.764	**0.997**	0.040	46.591	**0.998**	0.034	47.657	0.998	0.031	48.556	**0.999**	0.029
	S2NTNN-TV	**41.155**	**0.994**	**0.050**	**45.387**	**0.997**	**0.037**	**47.291**	**0.998**	**0.032**	**48.990**	**0.999**	**0.028**	**49.917**	**0.999**	**0.026**
HSI *Pavia* (200×200×80)	Observed	12.191	0.042	1.355	12.426	0.071	1.254	12.674	0.098	1.177	12.939	0.125	1.110	13.220	0.150	1.049
	TRLRF [65]	28.232	0.888	0.113	29.484	0.915	0.102	30.918	0.936	0.087	31.572	0.944	0.084	32.028	0.950	0.082
	TNN [1]	26.002	0.822	0.174	31.382	0.938	0.111	35.429	0.971	0.080	37.867	0.981	0.066	40.171	0.987	0.055
	FTNN [2]	32.345	0.954	0.079	37.821	0.985	0.052	42.066	0.992	0.039	45.266	0.996	0.030	48.447	0.997	0.024
	S2NTNN	38.755	0.990	0.027	46.164	0.998	0.016	50.803	**0.999**	**0.011**	52.021	**1.000**	0.010	53.075	**1.000**	0.009
	S2NTNN-TV	**38.837**	**0.993**	**0.026**	**47.825**	**0.999**	**0.013**	**50.994**	**0.999**	**0.011**	**52.741**	**1.000**	**0.009**	**54.381**	**1.000**	**0.008**
MSI *Balloons* (256×256×31)	Observed	13.529	0.205	1.389	13.762	0.248	1.278	14.010	0.286	1.194	14.272	0.320	1.123	14.554	0.350	1.059
	TRLRF [65]	30.062	0.883	0.244	34.450	0.952	0.167	38.868	0.982	0.112	39.907	0.985	0.103	40.288	0.986	0.101
	TNN [1]	26.321	0.850	0.267	34.521	0.961	0.161	38.822	0.982	0.111	41.355	0.990	0.087	43.253	0.993	0.071
	FTNN [2]	35.067	0.974	0.111	39.640	0.990	0.069	43.187	0.995	0.049	45.419	0.997	0.040	47.609	0.998	0.033
	S2NTNN	38.021	0.987	0.078	43.337	0.996	0.052	46.646	**0.998**	0.039	48.504	0.998	0.034	49.426	**0.999**	0.028
	S2NTNN-TV	**40.662**	**0.994**	**0.047**	**44.622**	**0.997**	**0.036**	**47.164**	**0.998**	**0.030**	**49.183**	**0.999**	**0.025**	**50.066**	**0.999**	**0.024**
MSI *Beads* (256×256×31)	Observed	14.414	0.187	1.406	14.646	0.227	1.295	14.899	0.267	1.211	15.165	0.309	1.139	15.438	0.349	1.073
	TRLRF [65]	18.010	0.449	0.688	23.255	0.738	0.476	26.211	0.845	0.356	31.150	0.948	0.218	32.259	0.958	0.197
	TNN [1]	19.976	0.584	0.580	23.284	0.773	0.434	26.004	0.866	0.344	28.283	0.916	0.278	30.230	0.944	0.230
	FTNN [2]	20.958	0.694	0.404	25.168	0.860	0.274	28.468	0.927	0.209	31.023	0.957	0.167	33.223	0.973	0.136
	S2NTNN	24.217	**0.846**	0.261	30.815	0.963	0.127	34.798	0.983	0.093	38.080	**0.991**	**0.072**	40.276	**0.994**	0.061
	S2NTNN-TV	**24.735**	0.834	**0.202**	**31.419**	**0.968**	**0.121**	**35.380**	**0.986**	**0.087**	**38.280**	**0.991**	0.074	**40.508**	**0.994**	**0.060**
MSI *Flowers* (256×256×31)	Observed	13.544	0.445	1.420	13.780	0.475	1.314	14.025	0.503	1.231	14.295	0.531	1.158	14.570	0.557	1.088
	TRLRF [65]	25.560	0.749	0.464	29.801	0.870	0.336	32.044	0.914	0.290	32.748	0.922	0.278	39.806	0.985	0.133
	TNN [1]	25.743	0.787	0.566	30.736	0.915	0.323	33.757	0.953	0.240	36.230	0.971	0.192	38.381	0.981	0.158
	FTNN [2]	29.411	0.918	0.218	34.014	0.965	0.147	36.899	0.980	0.116	39.317	0.988	0.094	41.501	0.992	0.079
	S2NTNN	31.564	0.938	0.266	36.997	0.978	0.138	41.424	0.991	0.075	44.175	0.995	0.055	46.596	0.997	0.050
	S2NTNN-TV	**32.430**	**0.961**	**0.147**	**38.339**	**0.988**	**0.064**	**41.914**	**0.994**	**0.050**	**44.309**	**0.996**	**0.042**	**46.849**	**0.998**	**0.035**

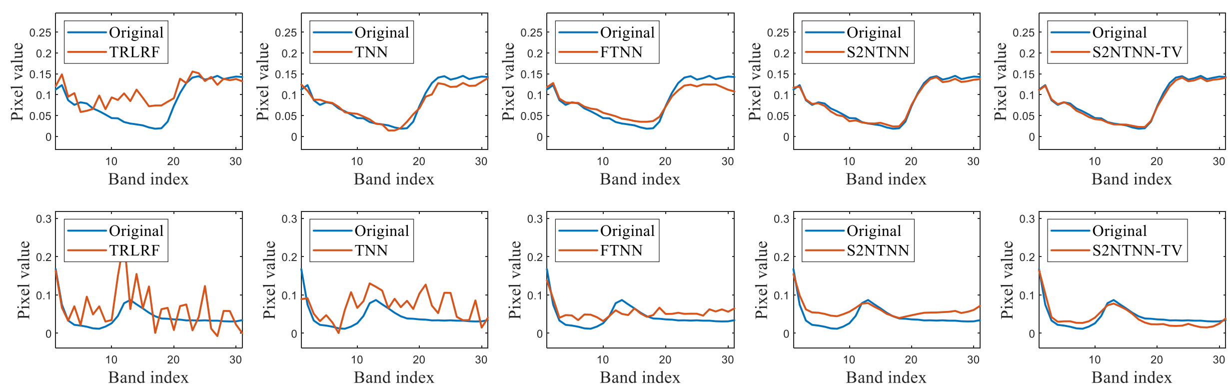


Fig. 4. The spectral curves of the tensor completion results by different methods on MSIs *Balloons* and MSI *Beads* with SR = 0.05.

For the observed data $\mathcal{O} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the total computational complexity of the ADMM algorithm at each iteration is $\mathcal{O}(2n_1n_2\tilde{n}_3(n_3 + \tilde{n}_3))$, where $\tilde{n}_3$ is the third dimension of the transformed tensor. More concretely, the computational complexity of the $\{\mathbf{\Theta}, \mathcal{X}\}$ sub-problem is $\mathcal{O}(2n_1n_2\tilde{n}_3(n_3 + \tilde{n}_3))$. The computational complexity of the $\mathcal{V}_d$ sub-problem is $\mathcal{O}(n_1n_2n_3)$. The computational complexity of updating Λ_d is $\mathcal{O}(n_1n_2n_3)$. Meanwhile, the number of parameters in the S2NT f and the inverse transform g is $2(\tilde{n}_3^2 + n_3\tilde{n}_3)$.

V. EXPERIMENTS

In this section, we introduce four multi-dimensional image recovery problems, i.e., tensor completion, background subtraction, RTC, and SCI. Each of these problems can be addressed using S2NTNN and S2NTNN-TV, where the only difference is the fidelity term $L(g(f(\mathcal{X})), \mathcal{O})$. We remark here that our method characterizes the low-rank structure of multi-dimensional images with compact representation abilities. Thus, it is not limited to the above applications. For other applications, e.g., multi-dimensional image denoising [46], super-resolution [38], and subspace clustering [30], with suitable formulations of the fidelity term, our method is believed to perform well.

A. Experimental Settings

In our method, the hyperparameters include the third dimension of the transformed tensor $\tilde{n}_3$, the Lagrange parameter β, the regularization parameters τ and λ, and the maximum iteration number t_{max}. Specifically, we set $\tilde{n}_3 = 2n_3$ and $\beta = 1$ for all tasks. We select τ and λ from the candidate

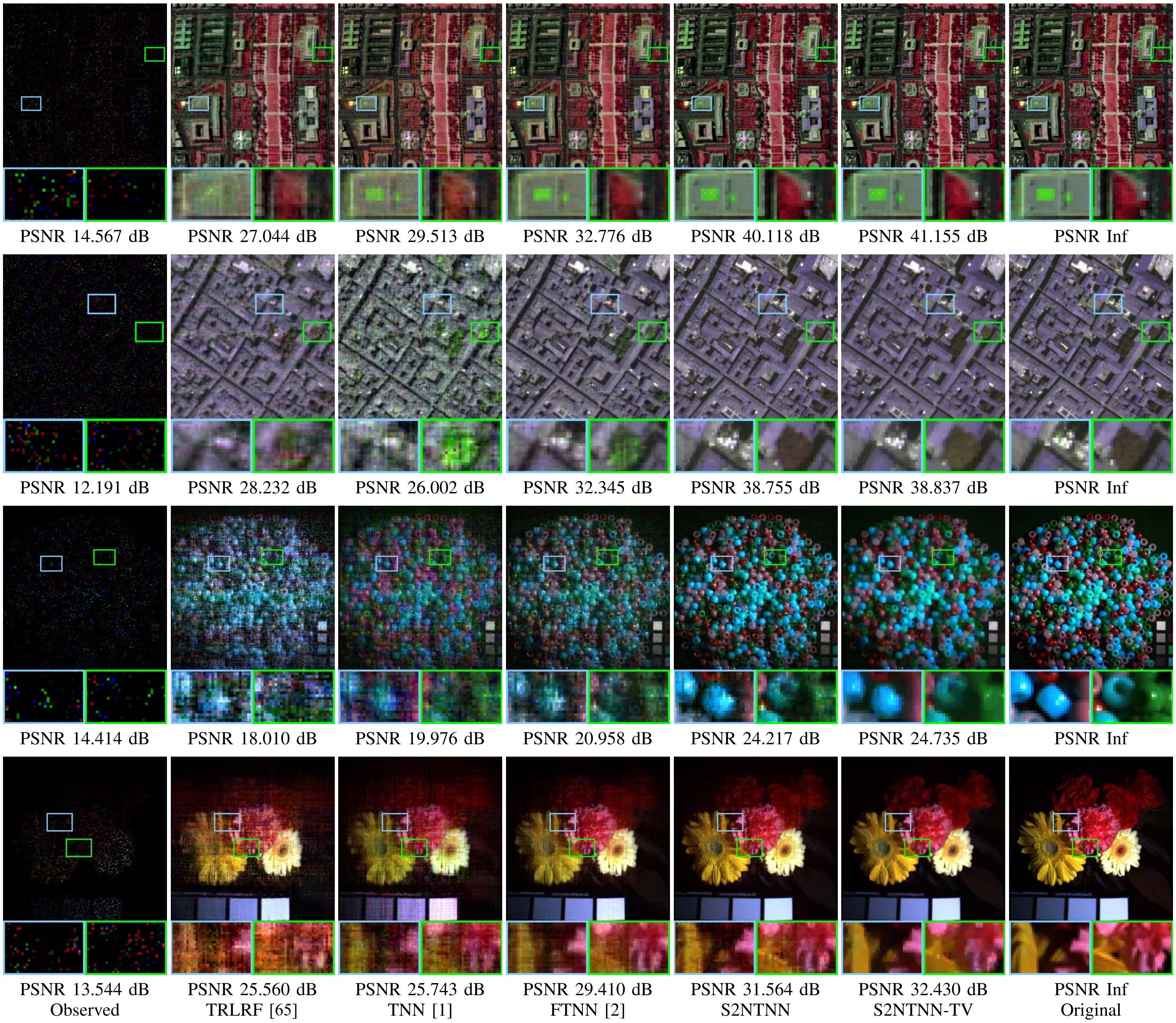


Fig. 5. The recovered results by different methods for tensor completion on HSI *WDC mall* (composed of the 50-th, 100-th, and the 150-th bands) with SR = 0.05, HSI *Pavia* (composed of the 1-st,10-th, and the 20-th bands) with SR = 0.05, MSI *Beads* (composed of the 10-th, 20-th, and the 30-th bands) with SR = 0.05, and MSI *Flowers* (composed of the 10-th, 20-th, and the 30-th bands) with SR = 0.05.

sets $\{10^{-j}\}_{j=1}^{8}$ and $\{10^{-j}\}_{j=2}^{5}$, respectively, to obtain the best PSNR value for all tasks. We set $t_{max} = 7000$, 1000, 7000, and 4000 for tensor completion, background subtraction, RTC, and SCI, respectively. The number of network layers is set to $p = q = 2$ for all tasks and the learning rate of the Adam optimizer is set to 0.005 for all tasks. The nonlinear activation function $\sigma(\cdot)$ is chosen as the LeakyReLU function with the negative slope 0.2 for all tasks. The size of the input data $\mathcal{X}$ is the same as the size of the observed data $\mathcal{O}$ for all tasks. We would like to emphasize that the proposed S2NT f and the inverse transform g are self-supervisedly learned by solely using the observed data. Thus, no training data and training/testing data splitting are required.

All experiments are conducted on the platform of Windows 10 with an Intel(R) Core i5-9400f CPU and RTX 2080 GPU with 24 GB RAM. Our method is implemented on PyTorch 1.9.0 with CPU and GPU calculation. All the compared methods are implemented on MATLAB R2019b with CPU calculation.

We use three numerical evaluation indices: peak signal to noise ratio (PSNR), structural similarity (SSIM), and spectral angle mapper (SAM) [66]. Higher PSNR and SSIM values correspond to better quality, while lower SAM value represents a smaller spectral angle between the ground truth and the recovered result.

B. Tensor Completion

The tensor completion [24], [67], [68] aims at recovering the original tensor from the incompleted tensor with random

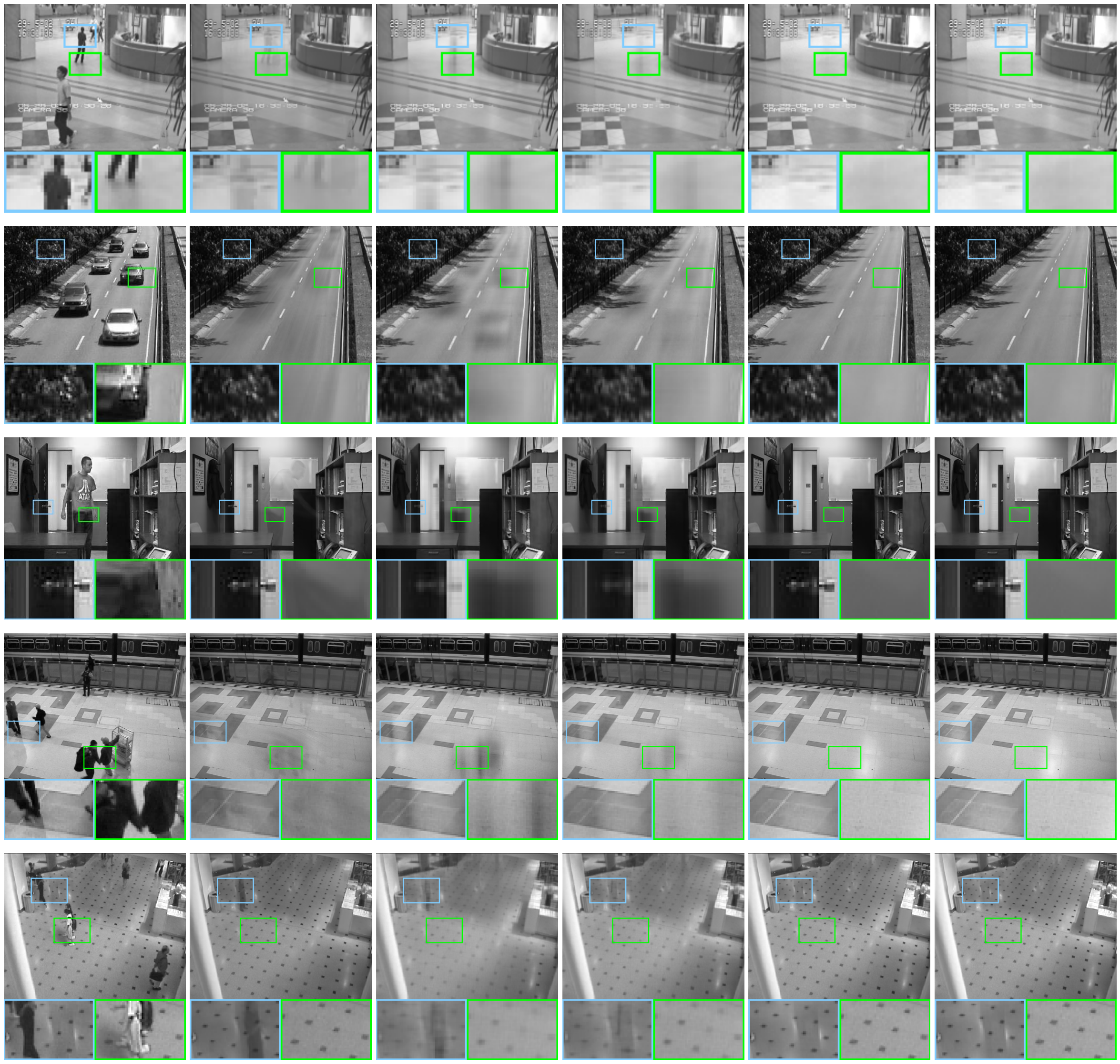


Fig. 6. The separated background by different methods for background subtraction on videos *Port* ($144 \times 176 \times 250$), *Highway* ($240 \times 320 \times 200$), *Office* ($240 \times 360 \times 200$), *PET* ($288 \times 360 \times 300$), and *Shop* ($256 \times 320 \times 100$).

sampling. The proposed S2NTNN model for tensor completion is formulated as

$$\min_{\mathbf{\Theta},\mathcal{X}} \lambda \sum_{k=1}^{\tilde{n}_3} \left\| (f(\mathcal{X}))^{(k)} \right\|_* + \|\mathcal{P}_\Omega(g(f(\mathcal{X})) - \mathcal{O})\|_F^2, \quad (21)$$

where $\mathcal{O}$ is the incompleted tensor, $\sum_{k=1}^{\tilde{n}_3} \left\| (f(\mathcal{X}))^{(k)} \right\|_*$ is the S2NTNN regularization, $\|\mathcal{P}_\Omega(g(f(\mathcal{X})) - \mathcal{O})\|_F^2$ is the fidelity term, and $\mathcal{P}_\Omega(\cdot)$ is the projection function that keeps the elements in the observed set $\mathbf{\Omega}$ and making others be zero. The final recovered result is $g(f(\mathcal{X}))$.

1) Datasets and Compared Methods: To illustrate the effectiveness of our method for tensor completion, we collected multi-dimensional image data including MSIs (*Balloons*, *Beads*, and *Flowers*[2] [71]) and HSIs (*Pavia* and *WDC mall*[3]). Five cases with sampling rates (SRs) $0.05, 0.1, 0.15, 0.2$, and 0.25 are established. The competing methods for tensor completion are: The tensor ring decomposition-based method TRLRF [65], the linear transform-based methods TNN (induced by DFT) [1] and FTNN (induced by framelet transform) [2]. The initialization function Init$(\cdot)$ for tensor completion is the linear interpolation that used in [49], which provides an ideal initialization with less time.

2) Experimental Results: The numerical results for tensor completion are illustrated in Table II. We can see that S2NTNN could achieve better PSNR and SSIM values than competing methods. Also, S2NTNN achieves better SAM values, which shows that S2NTNN preferably exploits the correlation along the third mode. We can observe that S2NTNN-TV generally has better performances than S2NTNN, which validates the effectiveness of the TV regularization to enhance the recovery performance.

[2]https://www.cs.columbia.edu/CAVE/databases/multispectral/
[3]https://engineering.purdue.edu/biehl/MultiSpec/hyperspectral.html

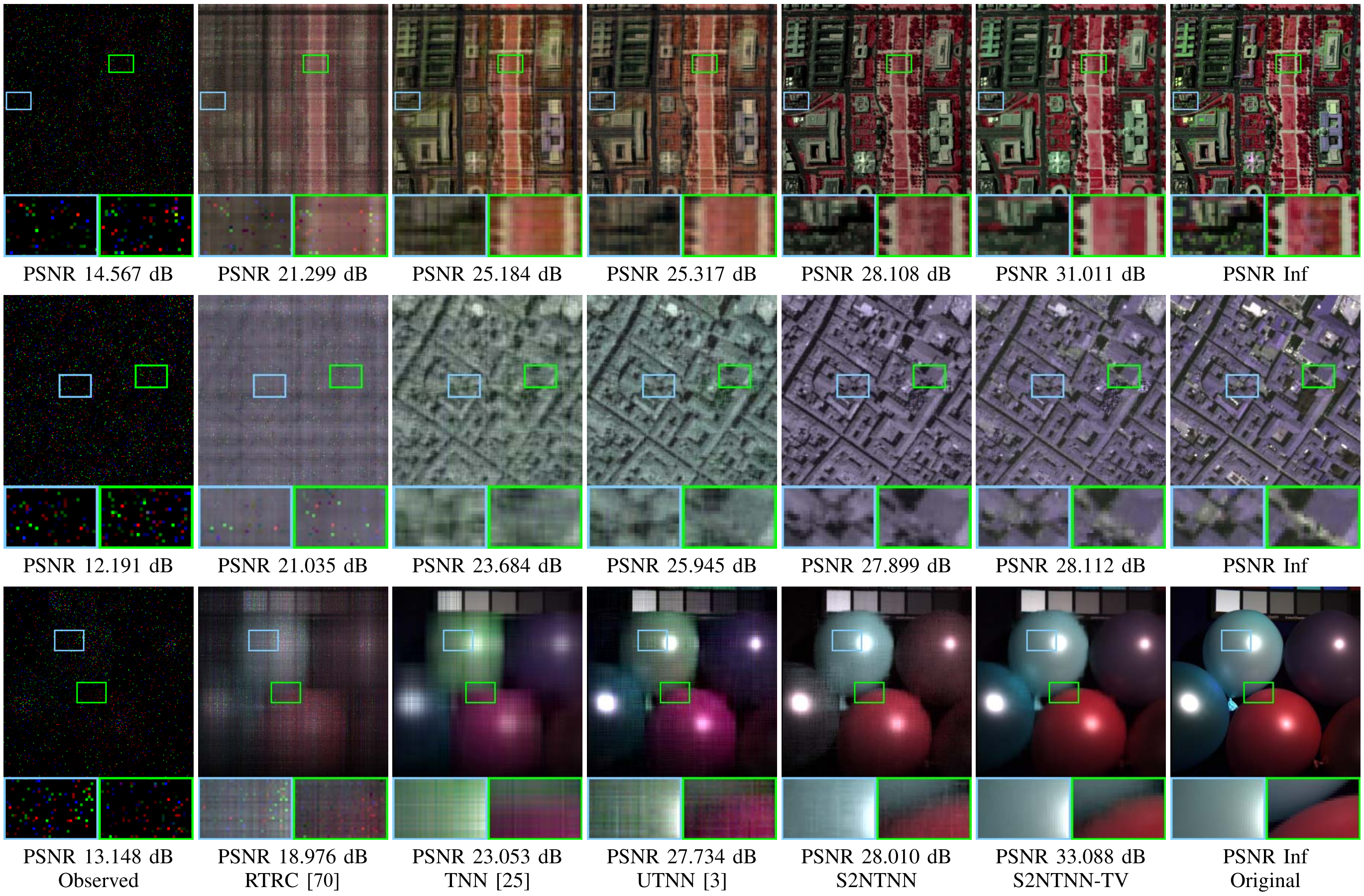


Fig. 7. The recovered results by different methods for RTC on HSI *WDC mall* (composed of the 50-th, 100-th, and the 150-th bands) with SR = 0.05, HSI *Pavia* (composed of the 1-st,10-th, and the 20-th bands) with SR = 0.05, and MSI *Balloons* (composed of the 10-th, 20-th, and the 30-th bands) with SR = 0.05.

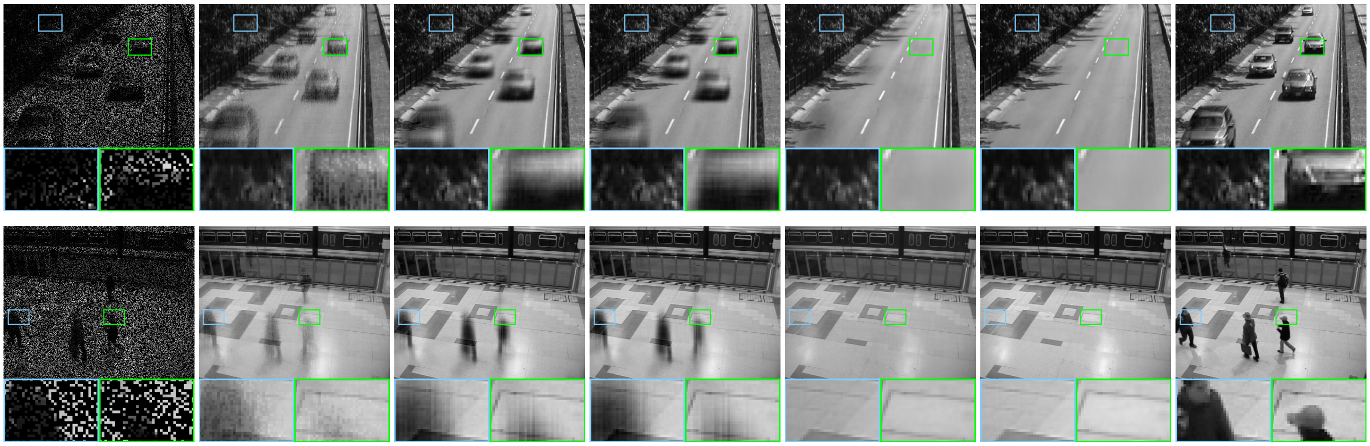
TABLE III

THE QUANTITATIVE RESULTS BY DIFFERENT METHODS ON DIFFERENT DATA FOR RTC. THE **BEST** VALUE ARE HIGHLIGHTED BY **BOLDFACE**, AND THE SECOND-BEST VALUE ARE HIGHLIGHTED BY UNDERLINED

Data	SR	0.05			0.1			0.15			0.2			0.25		
	Metric	PSNR	SSIM	SAM	PSNR	SSIM	SAM	PSNR	SSIM	SAM	PSNR	SSIM	SAM	PSNR	SSIM	SAM
HSI *WDC mall* (256×256×191)	Observed	13.953	0.066	1.405	13.626	0.088	1.351	13.351	0.105	1.309	13.109	0.117	1.271	12.903	0.126	1.236
	RTRC [70]	21.299	0.577	0.268	23.467	0.714	0.232	25.177	0.796	0.208	26.463	0.843	0.194	27.660	0.878	0.181
	TNN [25]	25.184	0.804	0.245	29.418	0.919	0.171	32.307	0.955	0.137	34.453	0.971	0.116	36.272	0.979	0.102
	UTNN [3]	25.317	0.818	0.218	30.816	0.949	0.126	34.890	0.979	0.086	37.898	**0.989**	**0.065**	40.572	**0.994**	**0.052**
	S2NTNN	28.108	0.921	0.158	33.356	0.972	0.106	34.641	0.983	0.081	37.172	0.988	0.075	38.689	0.991	0.062
	S2NTNN-TV	**31.011**	**0.952**	**0.107**	**35.493**	**0.983**	**0.076**	**36.364**	**0.986**	**0.080**	**40.565**	0.977	0.088	**45.062**	0.990	0.063
HSI *Pavia* (200×200×80)	Observed	11.941	0.035	1.383	11.918	0.055	1.310	11.894	0.069	1.259	11.874	0.080	1.215	11.859	0.090	1.176
	RTRC [70]	21.035	0.519	0.142	22.006	0.599	0.162	23.142	0.676	0.164	24.151	0.737	0.160	25.024	0.781	0.157
	TNN [25]	23.684	0.732	0.148	28.133	0.902	0.118	31.243	0.947	0.097	33.806	0.966	0.083	35.719	0.974	0.075
	UTNN [3]	25.946	0.844	0.124	30.758	0.945	0.089	33.633	0.968	0.073	35.662	0.977	0.065	36.820	0.982	0.060
	S2NTNN	27.901	0.916	0.105	32.629	0.965	0.049	37.491	0.989	0.033	40.318	0.994	0.029	41.750	**0.996**	0.025
	S2NTNN-TV	**28.112**	**0.920**	**0.078**	**33.588**	**0.967**	**0.042**	**39.086**	**0.992**	**0.029**	**42.018**	**0.996**	**0.023**	**42.406**	**0.996**	**0.019**
MSI *Balloons* (256×256×31)	Observed	13.148	0.169	1.411	12.998	0.171	1.324	12.860	0.168	1.268	12.716	0.158	1.228	12.578	0.149	1.196
	RTRC [70]	18.976	0.701	0.369	24.512	0.879	0.222	27.762	0.930	0.159	29.835	0.950	0.133	31.450	0.963	0.110
	TNN [25]	23.053	0.882	0.266	29.143	0.950	0.172	31.898	0.969	0.127	33.856	0.978	0.103	35.477	0.984	0.086
	UTNN [3]	27.734	0.890	0.281	31.010	0.907	0.232	32.700	0.971	0.126	35.950	0.979	0.098	37.667	0.986	0.062
	S2NTNN	28.010	0.921	0.256	32.837	0.972	0.163	35.732	0.984	0.109	37.543	0.990	0.083	39.272	0.994	0.054
	S2NTNN-TV	**33.088**	**0.975**	**0.136**	**36.495**	**0.988**	**0.069**	**39.465**	**0.994**	**0.051**	**40.041**	**0.995**	**0.050**	**41.228**	**0.996**	**0.047**

Some visual results for tensor completion are shown in Fig. 5. We can see that S2NTNN and S2NTNN-TV recover the images better than competing methods. S2NTNN-TV achieves better recovery in the spatial domain, especially according to the results on MSI *Beads*. This is due to the consideration of the spatial local smoothness of the TV regularization.

In addition, we plot the spectral curves of the recovered results in Fig. 4. S2NTNN and S2NTNN-TV more faithfully



Observed RTRC [70] TNN [25] UTNN [3] S2NTNN S2NTNN-TV Original

Fig. 8. The separated background by different methods for RTC on videos *highway* ($240 \times 320 \times 200$) and *PET* ($288 \times 360 \times 300$) with SR = 0.25.

TABLE IV

THE QUANTITATIVE RESULTS BY DIFFERENT METHODS ON DIFFERENT DATA FOR SCI. THE **BEST** VALUE ARE HIGHLIGHTED BY **BOLDFACE**, AND THE SECOND-BEST VALUE ARE HIGHLIGHTED BY UNDERLINED

Data	SR	0.05			0.1			0.15			0.2			0.25		
	Metric	PSNR	SSIM	SAM	PSNR	SSIM	SAM	PSNR	SSIM	SAM	PSNR	SSIM	SAM	PSNR	SSIM	SAM
MSI *Toys* ($256\times256\times31$)	GAP-TV [4]	21.488	0.642	0.774	22.318	0.691	0.744	22.692	0.732	0.699	22.817	0.755	0.667	22.766	0.772	0.648
	SeSCI [72]	20.815	0.612	0.590	21.574	0.689	0.602	21.668	0.722	0.602	21.471	0.738	0.602	21.218	0.749	0.603
	DeSCI [5]	19.702	0.624	0.410	21.211	0.735	0.426	22.148	0.785	0.437	22.871	0.812	0.413	23.220	0.828	0.409
	S2NTNN	23.876	0.792	0.504	24.927	0.830	0.458	25.311	0.840	0.508	25.940	0.862	0.494	26.464	0.872	0.508
	S2NTNN-TV	**24.209**	**0.803**	**0.434**	**25.424**	**0.847**	**0.436**	**26.308**	**0.863**	**0.470**	**26.791**	**0.880**	**0.450**	**27.122**	**0.885**	**0.486**
MSI *Flowers* ($256\times256\times31$)	GAP-TV [4]	22.944	0.655	0.732	24.024	0.702	0.683	24.585	0.741	0.633	24.864	0.766	0.597	25.121	0.782	0.577
	SeSCI [72]	22.405	0.658	0.551	23.947	0.725	0.546	24.417	0.758	0.537	24.578	0.777	0.531	24.657	0.786	0.535
	DeSCI [5]	21.150	0.633	0.465	22.872	0.737	0.411	23.927	0.783	0.402	24.604	0.810	0.390	24.693	0.826	0.382
	S2NTNN	26.253	0.812	0.479	26.860	0.852	0.613	28.505	**0.878**	0.589	28.573	0.884	**0.583**	29.314	0.894	**0.564**
	S2NTNN-TV	**26.464**	**0.839**	**0.291**	**27.558**	**0.857**	**0.605**	**28.602**	0.877	**0.587**	**28.955**	**0.885**	0.603	**29.462**	**0.895**	0.590
Video *Drop* ($256\times256\times10$)	GAP-TV [4]	23.324	0.732	0.111	24.077	0.712	0.109	24.495	0.714	0.106	24.750	0.718	0.104	25.248	0.737	0.098
	SeSCI [72]	24.171	0.850	0.070	26.135	0.869	0.066	27.029	0.878	0.065	27.430	0.883	0.064	27.921	0.888	0.062
	DeSCI [5]	21.551	0.806	0.061	22.880	0.813	0.064	24.348	0.843	0.067	25.169	0.860	0.066	26.283	0.877	0.065
	S2NTNN	25.028	0.881	**0.042**	26.024	0.859	**0.042**	26.378	0.869	**0.042**	27.422	0.883	**0.042**	27.664	0.889	**0.041**
	S2NTNN-TV	**26.674**	**0.894**	**0.042**	**27.519**	**0.898**	0.043	**27.396**	**0.892**	**0.042**	**27.637**	**0.908**	**0.042**	**27.951**	**0.913**	0.042
Video *Crash* ($256\times256\times10$)	GAP-TV [4]	20.557	0.636	0.265	21.171	0.626	0.267	21.546	0.630	0.261	21.806	0.642	0.258	22.083	0.661	0.252
	SeSCI [72]	20.016	0.698	0.203	21.301	0.699	0.207	21.880	0.716	0.204	22.126	0.722	0.206	22.345	0.734	0.205
	DeSCI [5]	19.821	0.718	0.152	20.378	0.708	0.177	21.068	0.727	0.180	21.178	0.732	0.189	21.305	0.746	0.193
	S2NTNN	21.469	0.787	0.126	22.117	0.706	0.137	22.781	0.780	**0.125**	22.993	0.806	0.137	23.527	0.829	0.126
	S2NTNN-TV	**21.906**	**0.790**	**0.125**	**22.901**	**0.796**	**0.126**	**23.201**	**0.821**	**0.125**	**23.403**	**0.830**	**0.126**	**23.598**	**0.842**	**0.125**

capture the nonlinear nature of spectral curves due to the nonlinear modeling capability of S2NT.

C. Background Subtraction

The background subtraction [2], [35], [69] aims at subtracting low-rank background from the original video. The proposed S2NTNN model for background subtraction is formulated as

$$\min_{\mathbf{\Theta},\mathcal{X}} \lambda \sum_{k=1}^{\tilde{n}_3} \left\| (f(\mathcal{X}))^{(k)} \right\|_* + \| g(f(\mathcal{X})) - \mathcal{O} \|_{\ell_1}, \tag{22}$$

where $\mathcal{O}$ is the original video, $\sum_{k=1}^{\tilde{n}_3} \left\| (f(\mathcal{X}))^{(k)} \right\|_*$ is the S2NTNN regularization, and $\| g(f(\mathcal{X})) - \mathcal{O} \|_{\ell_1}$ is the fidelity term. The low-rank background is obtained through $g(f(\mathcal{X}))$.

1) Datasets and Compared Methods: Five video frames[4] that contain low-rank background and sparse foreground are selected. The competing methods for the background subtraction are: The matrix robust principal component analysis method FastRPCA [69], the linear transform-based methods TNN (induced by DFT) [25] and DCTNN (induced by DCT) [47]. We directly use the original video as the initialization of $\mathcal{X}$ for background subtraction.

2) Experimental Results: The results by different methods for background subtraction are shown in Fig. 6. We can see that S2NTNN and S2NTNN-TV more precisely subtract the low-rank background. In addition, we can see from the zoom-in figures that S2NTNN and S2NTNN-TV more faithfully preserve the image details in the background than competing methods (e.g., the door handle in *Office* and the

[4] http://trace.eas.asu.edu/yuv/ and http://jacarini.dinf.usherbrooke.ca/static/dataset/

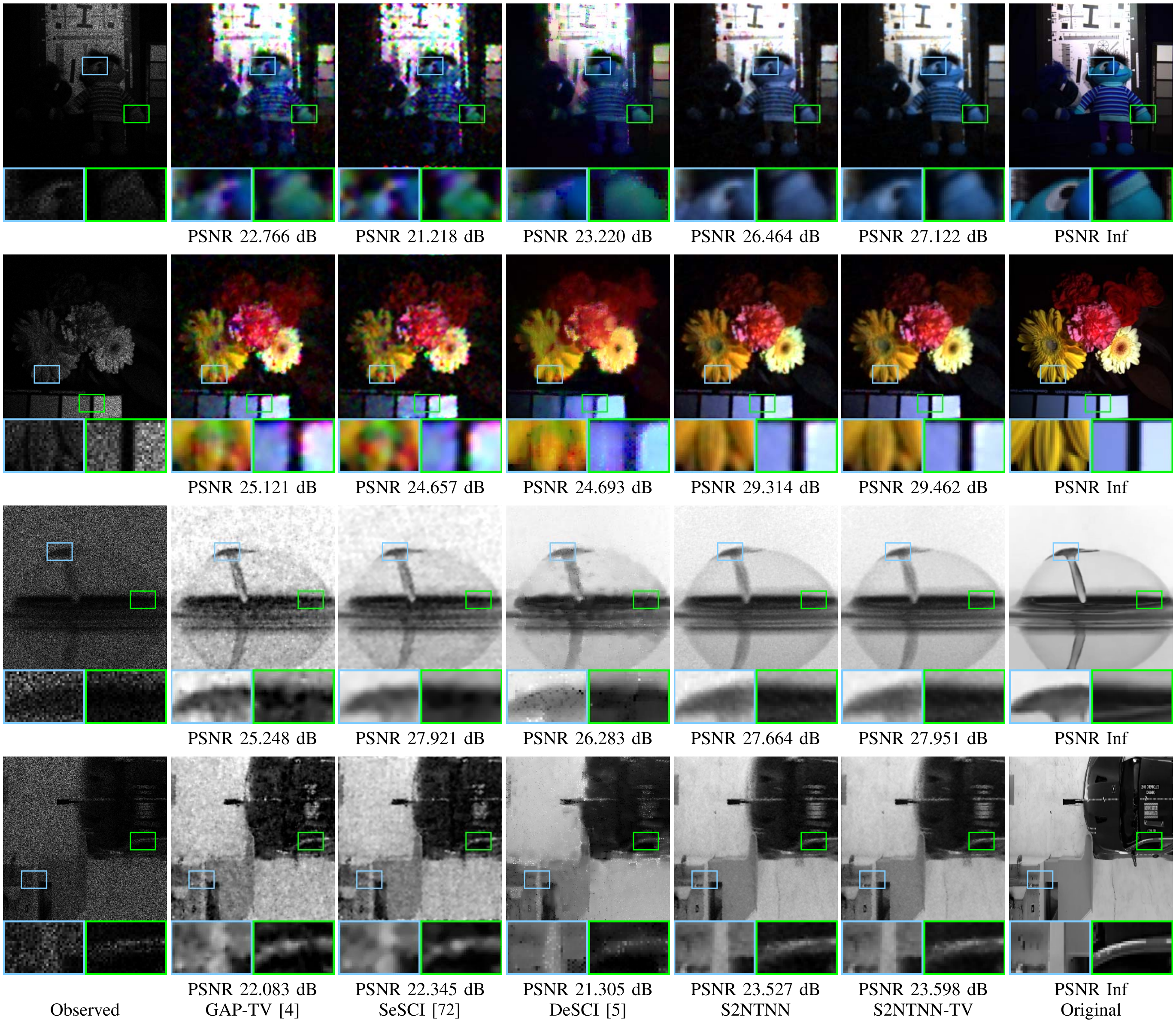


Fig. 9. The recovered results by different methods for SCI on MSI *Toys* (composed of the 10-th, 20-th, and the 30-th bands) with SR = 0.25, MSI *Flowers* (composed of the 10-th, 20-th, and the 30-th bands) with SR = 0.25, video *Drop* with SR = 0.25, and video *Crash* with SR = 0.25.

ground pattern in *Shop*). This can attribute to the nonlinear modeling ability of S2NT, which more compactly represents the low-rank tensor.

D. Robust Tensor Completion

The RTC [3], [70] aims at recovering the low-rank tensor from the incompleted tensor and simultaneously separate the sparse component. The proposed S2NTNN model for RTC is formulated as

$$\min_{\mathbf{\Theta},\mathcal{X}} \lambda \sum_{k=1}^{\tilde{n}_3} \left\|(f(\mathcal{X}))^{(k)}\right\|_* + \|\mathcal{P}_\Omega(g(f(\mathcal{X})) - \mathcal{O})\|_{\ell_1}, \tag{23}$$

where $\mathcal{O}$ is the incompleted tensor, $\sum_{k=1}^{\tilde{n}_3} \left\|(f(\mathcal{X}))^{(k)}\right\|_*$ is the S2NTNN regularization, and $\|\mathcal{P}_\Omega(g(f(\mathcal{X})) - \mathcal{O})\|_{\ell_1}$ is the fidelity term. The recovered result is $g(f(\mathcal{X}))$.

1) Datasets and Compared Methods: To illustrate the superiority of our method on RTC, we adopted the HSIs *Pavia* and *WDC mall*, the MSI *Balloons*, and the videos *Highway* and *PET* as the experimental data. For HSIs and MSI, we firstly sample the data using different SRs to obtain incompleted tensors, and then perform sparse noise degradation with noise sampling rate 0.1 on the incomplete data. For videos *Highway* and *PET*, we only sample the data using different SRs and do not perform the sparse noise. This is because the videos *Highway* and *PET* contain moving objects which act as the sparse foreground component. The RTC problem for videos *Highway* and *PET* aims to simultaneously infer the missing entries and separate the background and foreground. The competing methods for RTC are: The tensor ring decomposition-based method RTRC [70], the linear transform-based methods TNN (induced by DFT) [25] and UTNN (induced by unitary

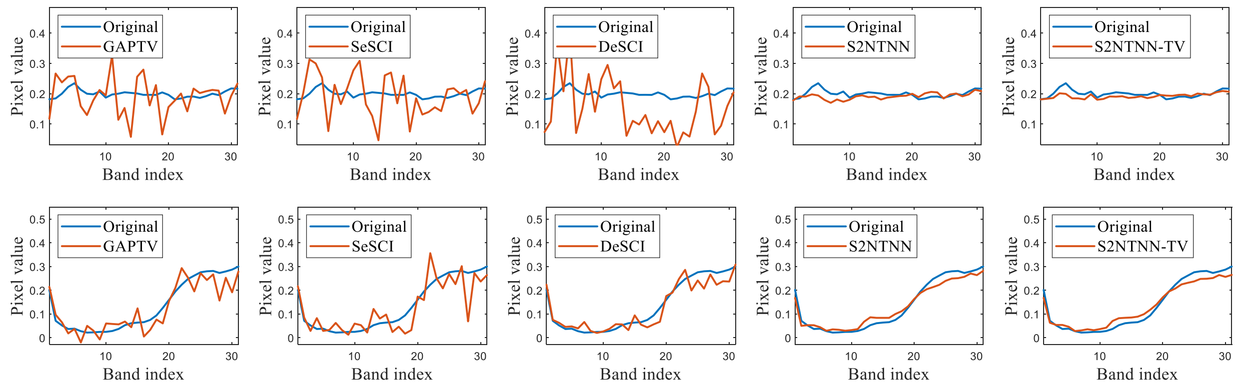


Fig. 10. The spectral curves of recovered results by different methods for SCI on MSIs *Toys* and *Flowers* with SR = 0.25.

TABLE V

THE QUANTITATIVE RESULTS FOR TENSOR COMPLETION ON MSI *Flowers* WITH SR = 0.1. S2NTNN (LINEAR) DENOTES THE S2NT f HAS NO NONLINEAR FUNCTION. S2NTNN (p) INDICATES THAT THE S2NT f HAS p NOFC$_3$ LAYERS. S2NTNN WO REG. DENOTES THE S2NTNN MODEL WITHOUT THE LOW-RANK REGULARIZATION

	Method	PSNR	SSIM	SAM
Nonlinearity	S2NTNN (Linear)	35.786	0.973	0.164
	S2NTNN (ReLU)	36.850	**0.980**	0.122
	S2NTNN (LeakyReLU)	**36.997**	0.978	0.138
	S2NTNN (PReLU)	36.734	0.977	0.134
	S2NTNN (PLU)	36.620	0.979	**0.106**
Hierarchy	S2NTNN (1)	36.434	0.975	0.139
	S2NTNN (2)	36.997	0.978	0.138
	S2NTNN (3)	**37.407**	**0.981**	**0.133**
	S2NTNN (4)	36.612	0.976	0.156
	S2NTNN (5)	35.863	0.969	0.203
	S2NTNN (10)	31.512	0.921	0.389
Regularizers	S2NTNN wo reg.	33.397	0.943	0.306
	S2NTNN (Low-rank)	**36.997**	**0.978**	**0.138**
	S2NTNN (Sparse)	34.179	0.961	0.284

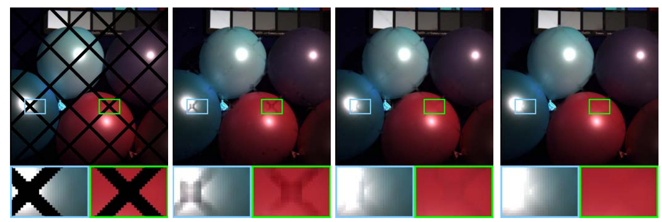
transform) [3]. We use the linear interpolation [49] as the initialization function Init(·) for S2NTNN and S2NTNN-TV.

2) Experimental Results: The numerical results for RTC are reported in Table III. We can see that S2NTNN-TV outperforms competing methods in terms of PSNR. However, UTNN achieves better SSIM and SAM values than S2NTNN-TV on HSI *WDC mall* with higher SRs. This is due to the consideration of spatial smoothness by S2NTNN-TV, where the over smoothness may influence the details preserving.

Some visual results for RTC are illustrated in Fig. 7 and Fig. 8. From Fig. 7, we can see that S2NTNN and S2NTNN-TV recover the tensor better than competing methods. S2NTNN-TV has smoother results than S2NTNN due to the TV regularization, which results in higher PSNR values. The separated background of videos *highway* and *PET* are shown in Fig. 8, where the original videos containing background and foreground are displayed as references. We can discover that the proposed methods have better performance for separating the low-rank background from the videos.

E. Snapshot Compressive Imaging

The SCI [5], [56], [73] is developed to capture multi-dimensional data from low-dimensional data with low



Observed 19.8dB FTNN [2] 33.9dB S2NTNN 38.1dB S2NTNN-TV 39.9dB

Fig. 11. The recovered results and corresponding PSNR values by different methods for tensor completion on MSI *Balloons* with structure missing.

computational cost by summing up the spectral/temporal signals to obtain the measurement. The key module in the SCI system is the reconstruction of the original high-dimensional signals. Given the observed measurement $\mathcal{O} \in \mathbb{R}^{n_1 \times n_2 \times 1}$, the proposed S2NTNN model for SCI is formulated as

$$\min_{\Theta,\mathcal{X}} \lambda \sum_{k=1}^{\tilde{n}_3} \left\| (f(\mathcal{X}))^{(k)} \right\|_* + \left\| \sum_{k=1}^{n_3} \mathcal{C}^{(k)} \odot (g(f(\mathcal{X})))^{(k)} - \mathcal{O} \right\|_F^2, \tag{24}$$

where $(g(f(\mathcal{X})))^{(k)}$ and $\mathcal{C}^{(k)}$ respectively denote the k-th frontal slice of the underlying low-rank tensor and the given mask. $\odot$ denotes the element-wise product. Here, $\sum_{k=1}^{\tilde{n}_3} \left\| (f(\mathcal{X}))^{(k)} \right\|_*$ is the S2NTNN regularization and $\left\| \sum_{k=1}^{n_3} \mathcal{C}^{(k)} \odot (g(f(\mathcal{X})))^{(k)} - \mathcal{O} \right\|_F^2$ is the fidelity term. The recovered result is $g(f(\mathcal{X}))$.

1) Datasets and Compared Methods: We adopted MSI *Toys*, MSI *Flowers*, video *Drop*, and video *Crash*[5] as the experimental data for SCI. We firstly sample the data using different SRs and then sum up the frontal slices to generate the sensing measurement. Gaussian noise with the standard deviation 0.1 is performed on the sensing measurement. The competing methods for SCI are: The TV-based method GAP-TV [4], the low-rankness-based method DeSCI [5], and the sparsity-based method SeSCI [72]. We use the recovered results of GAP-TV as the initialization of DeSCI, S2NTNN, and S2NTNN-TV.

[5]https://drive.google.com/drive/folders/1d2uh9nuOL5Z7WnEQJ5HZSDMWK2VAT9sH

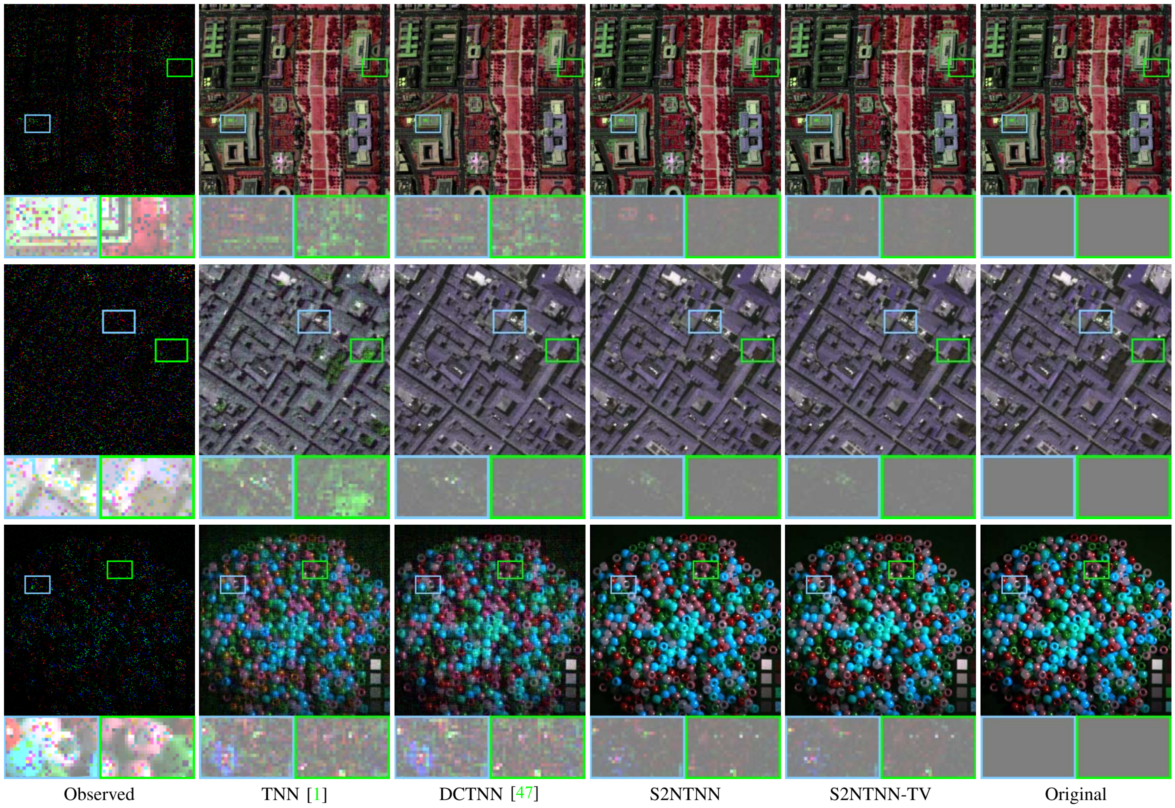


Fig. 12. The recovered results by different methods for tensor completion on HSI *WDC mall*, HSI *Pavia*, and MSI *Beads* with SR = 0.1. We display the residual images (difference between the ground truth and the recovered result) in zoom-in figures. Residual images with less color information indicate better performance.

2) Experimental Results: The numerical results for SCI are shown in Table IV. We can see that S2NTNN-TV outperforms competing methods with a considerable margin. The visual results for SCI are illustrated in Fig. 9. We can see that the proposed S2NTNN and S2NTNN-TV can recover the images more precisely. Moreover, we plot the spectral curves of the recovered results for SCI in Fig. 10. We can see that S2NTNN and S2NTNN-TV preserve the nonlinear spectral curves better than other methods due to the nonlinear modeling capability of S2NT.

F. Discussions

1) Compact Representation by S2NT: To demonstrate that the proposed S2NT can obtain a better low-rank representation than linear transforms, we plot the AccEgy [31] with respect to the percentage of singular values of the transformed frontal slices in Fig. 2. The transformed frontal slices are obtained by S2NT, S2NT (Linear, 1), DCT, and DFT. Here, S2NT (Linear, 1) denotes that the S2NT f only have one linear layer without nonlinear activation function. We can observe that S2NT obtains a more compact representation with more energy concentrated in larger singular values. This can improve the recovery performance, where the data can be approximated via lower-rank representation. In contrast, S2NT (Linear, 1) obtains a less compact representation. This verifies the effectiveness of nonlinearity and the hierarchical structure of S2NT for obtaining a better low-rank representation.

Moreover, we display the recovered results and their residual images (difference between the ground truth and the recovered result) of TNN (induced by DFT) [1], DCTNN (induced by DCT) [47], and the proposed S2NT-based methods in Fig. 12. We can observe that the proposed methods achieve better details preservation and color preservation compared with linear transform-based methods due to the better low-rank representation.

2) Effectiveness of Nonlinearity: This section tests the influence of nonlinearity in the proposed method. Specifically, we compare the performance of S2NTNN without nonlinear layers (denoted as S2NTNN (Linear)) and S2NTNN with different nonlinear activation layers, i.e., ReLU, LeakyReLU, PReLU [59], and piecewise linear unit (PLU) [74]. The results are shown in the first block of Table V. We can see that the performance is considerably increased with nonlinear layers. This is because the nonlinear modeling ability could help to obtain a better low-rank representation.

3) Effectiveness of Hierarchy: In this section, we test the influence of the hierarchy, i.e., the number of layers of the proposed S2NT. Specifically, we change the number of NoFC_3 layers in the S2NT (i.e., the parameter p) to clarify the influence. The results are shown in the second block of Table V. When p is small, increasing p can enhance the performance. However, when p is larger, the results are not as desirable as we expected. The is because a deeper network is more likely to suffer from the vanishing gradient.

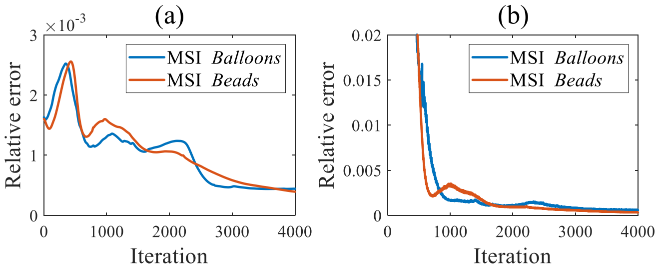


Fig. 13. The relative error with respect to iterations for tensor completion with SR = 0.25. (a) The relative error of f and g parameters, i.e., $\sum_{i=1}^{p+q} \left\|\mathbf{W}_i^{t+1} - \mathbf{W}_i^t\right\|_F^2 / \|\mathbf{W}_i^t\|_F^2$. (b) The relative error of $\mathcal{V}_d$, i.e., $\sum_{d=1,2} \left\|\mathcal{V}_d^{t+1} - \mathcal{V}_d^t\right\|_F^2 / \|\mathcal{V}_d^t\|_F^2$.

4) Low-Rankness vs Sparsity: The sparse modeling of the data has achieved great success [75]–[78]. Dose the sparsity works in our method? To clarify this, we replace the low-rank term with the sparse term, i.e., $\mathcal{L}_1 = \lambda \sum_{k=1}^{\tilde{n}_3} \left\|(f(\mathcal{O}))^{(k)}\right\|_{\ell_1}$, where ℓ_1-norm is the relaxation of ℓ_0-norm. Meanwhile, we use the S2NTNN without regularization (i.e., $\mathcal{L}_1 = 0$) as the baseline. The results are shown in the third block of Table V. We can observe that S2NTNN (Low-rank) outperforms S2NTNN (Sparse), which reveals that low-rankness is more effective to represent the third-order tensor in our method.

5) Effectiveness of TV Regularization: The S2NTNN only considers low-rankness of tensor data, which is limited to capture the spatial local similarity. This motivates us to perform the TV regularization on the spatial domain to faithfully explore the spatial local smoothness for better performance. To clarify this, we conduct the experiment for tensor completion where the incompleted entries are structurally sampled. The results are shown in Fig. 11. We can observe that S2NTNN-TV recovers the spatial information better than S2NTNN, which verifies the effectiveness of the TV regularization.

6) Convergence Analysis: To test the convergence behavior of the ADMM Algorithm 1, we plot the relative error of variables with respect to the iteration number in Fig. 13. The downward trend of the curves verifies the convergence behavior of our method.

VI. Conclusion

This paper suggests the S2NT-based TNN for multi-dimensional image recovery. The proposed S2NT obtains a better low-rank representation than that of linear transforms, which improves the recovery quality. We further introduce the TV regularization in the S2NTNN model and apply the ADMM algorithm to tackle the S2NTNN-TV model. Extensive experiments on different data for tensor completion, background subtraction, RTC, and SCI demonstrate the wide applicability of the proposed method and its superiority over state-of-the-art methods. In future work, we can consider extending our method to more applications such as denoising [46], super-resolution [38], and subspace clustering [30]. It is also interesting to extend our method to a weighted version [8] to further enhance the capability of our method.

References

[1] 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.

[2] 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.

[3] 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, p. e2299, May 2020.

[4] X. Yuan, "Generalized alternating projection based total variation minimization for compressive sensing," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 2539–2543.

[5] 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.

[6] Q. Xie *et al.*, "Multispectral images denoising by intrinsic tensor sparsity regularization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1692–1700.

[7] Z. Gu, F. Li, F. Fang, and G. Zhang, "A novel retinex-based fractional-order variational model for images with severely low light," *IEEE Trans. Image Process.*, vol. 29, pp. 3239–3253, 2020.

[8] Y. Xie, Y. Qu, D. Tao, W. Wu, Q. Yuan, and W. Zhang, "Hyperspectral image restoration via iteratively regularized weighted schatten p-norm minimization," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4642–4659, Apr. 2016.

[9] Y. Xie, S. Gu, Y. Liu, W. Zuo, W. Zhang, and L. Zhang, "Weighted schatten p-norm minimization for image denoising and background subtraction," *IEEE Trans. Image Process.*, vol. 25, no. 10, pp. 4842–4857, Aug. 2016.

[10] F. Fang, J. Li, and T. Zeng, "Soft-edge assisted network for single image super-resolution," *IEEE Trans. Image Process.*, vol. 29, pp. 4656–4668, 2020.

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

[12] B. Yaman, S. Weingartner, N. Kargas, N. D. Sidiropoulos, and M. Akcakaya, "Low-rank tensor models for improved multidimensional MRI: Application to dynamic cardiac T_1 mapping," *IEEE Trans. Comput. Imag.*, vol. 6, pp. 194–207, 2020.

[13] S. Li, R. Dian, L. Fang, and J. M. Bioucas-Dias, "Fusing hyperspectral and multispectral images via coupled sparse tensor factorization," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 4118–4130, Aug. 2018.

[14] C. Zhang, H. Fu, J. Wang, W. Li, X. Cao, and Q. Hu, "Tensorized multi-view subspace representation learning," *Int. J. Comput. Vis.*, vol. 128, no. 8, pp. 2344–2361, 2020.

[15] X. Guo and Y. Ma, "Generalized tensor total variation minimization for visual data recovery?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3603–3611.

[16] Y. Xie, D. Tao, W. Zhang, L. Zhang, and Y. Qu, "On unifying multi-view self-representations for clustering by tensor multi-rank minimization," *Int. J. Comput. Vis.*, vol. 126, no. 11, pp. 1157–1179, 2018.

[17] W. Hu, Y. Yang, W. Zhang, and Y. Xie, "Moving object detection using tensor-based low-rank and saliently fused-sparse decomposition," *IEEE Trans. Image Process.*, vol. 26, no. 2, pp. 724–737, Feb. 2017.

[18] H. Zhang, X.-L. Zhao, T.-X. Jiang, M. K. Ng, and T.-Z. Huang, "Multiscale feature tensor train rank minimization for multidimensional image recovery," *IEEE Trans. Cybern.*, early access, Sep. 20, 2021, doi: 10.1109/TCYB.2021.3108847.

[19] Y.-C. Miao, X.-L. Zhao, X. Fu, J.-L. Wang, and Y.-B. Zheng, "Hyperspectral denoising using unsupervised disentangled spatiospectral deep priors," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–16, 2021, doi: 10.1109/TGRS.2021.3106380.

[20] X. Liu, H. Shi, X. Hong, H. Chen, D. Tao, and G. Zhao, "3D skeletal gesture recognition via hidden states exploration," *IEEE Trans. Image Process.*, vol. 29, pp. 4583–4597, 2020.

[21] X. Liu and G. Zhao, "3D skeletal gesture recognition via discriminative coding on time-warping invariant Riemannian trajectories," *IEEE Trans. Multimedia*, vol. 23, pp. 1841–1854, 2021.

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

[23] W. Hu, D. Tao, W. Zhang, Y. Xie, and Y. Yang, "The twist tensor nuclear norm for video completion," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 12, pp. 2961–2973, Dec. 2017.

[24] 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.

[25] 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.

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

[27] C. Li, W. He, L. Yuan, Z. Sun, and Q. Zhao, "Guaranteed matrix completion under multiple linear transformations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11128–11137.

[28] H. Zeng, Y. Chen, X. Xie, and J. Ning, "Enhanced nonconvex low-rank approximation of tensor multi-modes for tensor completion," *IEEE Trans. Comput. Imag.*, vol. 7, pp. 164–177, 2021.

[29] X. Cao, Q. Zhao, D. Meng, Y. Chen, and Z. Xu, "Robust low-rank matrix factorization under general mixture noise distributions," *IEEE Trans. Image Process.*, vol. 25, no. 10, pp. 4677–4690, Oct. 2016.

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

[31] 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.

[32] J. Hou, F. Zhang, H. Qiu, J. Wang, Y. Wang, and D. Meng, "Robust low-tubal-rank tensor recovery from binary measurements," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Mar. 3, 2021, doi: 10.1109/TPAMI.2021.3063527.

[33] F. Zhang, J. Wang, W. Wang, and C. Xu, "Low-tubal-rank plus sparse tensor recovery with prior subspace information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3492–3507, Oct. 2021.

[34] Z. Zha, B. Wen, X. Yuan, J. T. Zhou, J. Zhou, and C. Zhu, "Triply complementary priors for image restoration," *IEEE Trans. Image Process.*, vol. 30, pp. 5819–5834, 2021.

[35] X. Liu, G. Zhao, J. Yao, and C. Qi, "Background subtraction based on low-rank and structured sparse decomposition," *IEEE Trans. Image Process.*, vol. 24, no. 8, pp. 2502–2514, Aug. 2015.

[36] Z. Zha, X. Yuan, B. Wen, J. Zhou, J. Zhang, and C. Zhu, "From rank estimation to rank approximation: Rank residual constraint for image restoration," *IEEE Trans. Image Process.*, vol. 29, pp. 3254–3269, 2020.

[37] Z. Zha, B. Wen, X. Yuan, J. Zhou, and C. Zhu, "Image restoration via reconciliation of group sparsity and low-rank models," *IEEE Trans. Image Process.*, vol. 30, pp. 5223–5238, 2021.

[38] R. Dian and S. Li, "Hyperspectral image super-resolution via subspace-based low tensor multi-rank regularization," *IEEE Trans. Image Process.*, vol. 28, no. 10, pp. 5135–5146, Oct. 2019.

[39] B. Romera-Paredes and M. Pontil, "A new convex relaxation for tensor completion," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2013, pp. 2967–2975.

[40] X.-Y. Liu, S. Aeron, V. Aggarwal, and X. Wang, "Low-tubal-rank tensor completion using alternating minimization," *IEEE Trans. Inf. Theory*, vol. 66, no. 3, pp. 1714–1737, Mar. 2020.

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

[42] X. Xiao, Y. Chen, Y.-J. Gong, and Y. Zhou, "Low-rank preserving t-Linear projection for robust image feature extraction," *IEEE Trans. Image Process.*, vol. 30, pp. 108–120, 2021.

[43] Y. Xu, Z. Wu, J. Chanussot, and Z. Wei, "Nonlocal patch tensor sparse representation for hyperspectral image super-resolution," *IEEE Trans. Image Process.*, vol. 28, no. 6, pp. 3034–3047, Jun. 2019.

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

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

[46] M. Baburaj and S. N. George, "Dct based weighted adaptive multi-linear data completion and denoising," *Neurocomputing*, vol. 318, pp. 120–136, Nov. 2018.

[47] 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.

[48] 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, Jul. 2021.

[49] 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.*, early access, Aug. 31, 2021, doi: 10.1109/TNNLS.2021.3104837.

[50] M. K. Ng, X. Zhang, and X.-L. Zhao, "Patched-tube unitary transform for robust tensor completion," *Pattern Recognit.*, vol. 100, Apr. 2020, Art. no. 107181.

[51] A. Kratsios, "The universal approximation property: Characterization, construction, representation, and existence," *Ann. Math. Artif. Intell.*, vol. 89, nos. 5–6, pp. 435–469, Jun. 2021.

[52] S. Arora, N. Cohen, W. Hu, and Y. Luo, "Implicit regularization in deep matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–12.

[53] J. Fan and T. W. Chow, "Non-linear matrix completion," *Pattern Recognit.*, vol. 77, pp. 378–394, May 2018.

[54] J. Fan and J. Cheng, "Matrix completion by deep matrix factorization," *Neural Netw.*, vol. 98, pp. 34–41, Feb. 2017.

[55] Z. Li, Z.-Q. John Xu, T. Luo, and H. Wang, "A regularized deep matrix factorized model of matrix completion for image restoration," 2020, *arXiv:2007.14581*.

[56] J. Ma, X.-Y. Liu, Z. Shou, and X. Yuan, "Deep tensor ADMM-net for snapshot compressive imaging," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 10222–10231.

[57] H. Chen and J. Li, "Neural tensor model for learning multi-aspect factors in recommender systems," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, Jul. 2020, pp. 2449–2455.

[58] S. Zhang, L. Wang, L. Zhang, and H. Huang, "Learning tensor low-rank prior for hyperspectral image reconstruction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 12001–12010.

[59] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.

[60] G. A. Watson, "Characterization of the subdifferential of some matrix norms," *Linear Algebra Appl.*, vol. 170, pp. 33–45, Jun. 1992.

[61] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–15.

[62] P. Cascarano, A. Sebastiani, M. C. Comes, G. Franchini, and F. Porta, "Combining weighted total variation and deep image prior for natural and medical image restoration via ADMM," 2020, *arXiv:2009.11380*.

[63] Y.-S. Luo, X.-L. Zhao, T.-X. Jiang, Y.-B. Zheng, and Y. Chang, "Hyperspectral mixed noise removal via spatial–spectral constrained unsupervised deep image prior," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 9435–9449, 2021.

[64] J.-H. Zhuang, Y.-S. Luo, X.-L. Zhao, and T.-X. Jiang, "Reconciling hand-crafted and self-supervised deep priors for video directional rain streaks removal," *IEEE Signal Process. Lett.*, vol. 28, pp. 2147–2151, 2021.

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

[66] S. Erturk, "Fuzzy fusion of change vector analysis and spectral angle mapper for hyperspectral change detection," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2018, pp. 5045–5048.

[67] X. Zhang and M. K.-P. Ng, "Low rank tensor completion with Poisson observations," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Feb. 15, 2021, doi: 10.1109/TPAMI.2021.3059299.

[68] P. Zhou, C. Lu, Z. Lin, and C. Zhang, "Tensor factorization for low-rank tensor completion," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1152–1163, Mar. 2018.

[69] A. Aravkin, S. Becker, V. Cevher, and P. Olsen, "A variational approach to stable principal component pursuit," in *Proc. Conf. Uncertainty Artif. Intell.*, 2014, pp. 1–10.

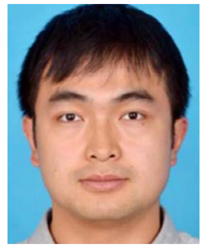
[70] H. Huang, Y. Liu, Z. Long, and C. Zhu, "Robust low-rank tensor ring completion," *IEEE Trans. Comput. Imag.*, vol. 6, pp. 1117–1126, 2020.

[71] F. Yasuma, T. Mitsunaga, D. Iso, and S. K. Nayar, "Generalized assorted pixel camera: Postcapture control of resolution, dynamic range, and spectrum," *IEEE Trans. Image Process.*, vol. 19, no. 9, pp. 2241–2253, Sep. 2010.

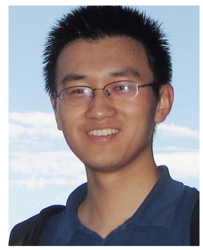
[72] P. Yang, L. Kong, X.-Y. Liu, X. Yuan, and G. Chen, "Shearlet enhanced snapshot compressive imaging," *IEEE Trans. Image Process.*, vol. 29, pp. 6466–6481, 2020.
[73] D. L. Donoho, "Compressed Sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Jan. 2006.
[74] A. Nicolae, "PLU: The piecewise linear unit activation function," 2018, *arXiv:1809.09534*.
[75] J. Deng, Z. Zhang, E. Marchi, and B. Schuller, "Sparse autoencoder-based feature transfer learning for speech emotion recognition," in *Proc. Humaine Assoc. Conf. Affect. Comput. Intell. Interact.*, Sep. 2013, pp. 511–516.
[76] 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.
[77] J. Yao, D. Meng, Q. Zhao, W. Cao, and Z. Xu, "Nonconvex-sparsity and nonlocal-smoothness-based blind hyperspectral unmixing," *IEEE Trans. Image Process.*, vol. 28, no. 6, pp. 2991–3006, Jun. 2019.
[78] Z. Zha *et al.*, "Group sparsity residual constraint for image denoising with external nonlocal self-similarity prior," *Neurocomputing*, vol. 275, pp. 2294–2306, Jan. 2018.

Yi-Si Luo is currently pursuing the bachelor's degree with the School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, China. His research interests include tensor modeling and image processing.

Xi-Le Zhao (Member, IEEE) received the M.S. and Ph.D. degrees from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2009 and 2012, respectively. He worked as a Postdoctoral Researcher with Prof. Michael K. Ng at Hong Kong Baptist University from 2013 to 2014. He worked as a Visiting Scholar with Prof. Jose Bioucas Dias at the University of Lisbon from 2016 to 2017. He is currently a Professor with the School of Mathematical Sciences, UESTC. His research interests include models, algorithms, and theories for the low-level inverse problems of multi-dimensional images. More information can be found on his homepage at: https://zhaoxile.github.io/



Tai-Xiang Jiang (Member, IEEE) received the Ph.D. degree in mathematics from the University of Electronic Science and Technology of China (UESTC) in 2019. He was a co-training Ph.D. Student with the University of Lisbon supervised by Prof. Jose M. Bioucas-Dias from 2017 to 2018. He was a Research Assistant with Hong Kong Baptist University supported by Prof. Michael K. Ng in 2019. 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 and tensor decomposition for multi-dimensional image processing, especially on the low-level inverse problems for multi-dimensional images. More details about him are available at: https://taixiangjiang.github.io/



Yi Chang (Member, IEEE) received the B.S. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2011, and the M.S. and Ph.D. degrees from the Huazhong University of Science and Technology in 2014 and 2019, respectively. He is currently a Lecturer with the School of Artificial Intelligence and Automation, Huazhong University of Science and Technology. His current research interests include adverse weather image restoration and computer vision.

Michael K. Ng (Senior Member, IEEE) received the B.Sc. and M.Phil. degrees from The University of Hong Kong, Hong Kong, in 1990 and 1992, respectively, and the Ph.D. degree from The Chinese University of Hong Kong, Hong Kong, in 1995. From 1995 to 1997, he was a Research Fellow with the Computer Sciences Laboratory, The Australian National University, Canberra, ACT, Australia. He was an Assistant Professor/an Associate Professor with The University of Hong Kong from 1997 to 2005. From 2006 to 2019, he was a Professor/a Chair Professor with the Department of Mathematics, Hong Kong Baptist University, Hong Kong. He is currently a Chair Professor with the Research Division of Mathematical and Statistical Science, The University of Hong Kong. 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 received the Feng Kang Prize for his significant contributions to scientific computing. He serves as an editorial board member for several international journals.

Chao Li received the bachelor's and Ph.D. degrees from Harbin Engineering University, Harbin, China, in 2010 and 2017, respectively. He was a Postdoctoral Researcher with the Center for Advanced Intelligence Project, RIKEN, Tokyo, Japan, from 2018 to 2020. He has been an Indefinite-Term Research Scientist with RIKEN since 2021. His research interests include multilinear algebra and statistical machine learning.