Neurocomputing 617 (2025) 129036

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Learning a more compact representation for low-rank tensor completion

Xi-Zhuo Li, Tai-Xiang Jiang, Liqiao Yang *, Guisong Liu

School of Computing and Artificial Intelligence, Southwestern University of Finance and Economics, Chengdu, Sichuan, PR China
Kash Institute of Electronics and Information Industry, Kashi, PR China
Engineering Research Center of Intelligent Finance, Ministry of Education, Southwestern University of Finance and Economics, Chengdu, Sichuan, PR China

ARTICLE INFO

Communicated by Y. Yang

Keywords:
Tensor completion
Tensor singular value decomposition
Nonlinear transform
Convolution
Multi-dimensional image

ABSTRACT

Transform-based tensor nuclear norm (TNN) methods have gained considerable attention for their effectiveness in addressing tensor recovery challenges. The integration of deep neural networks as nonlinear transforms has been shown to significantly enhance their performance. Minimizing transform-based TNN is equivalent to minimizing the ℓ_1 norm of singular values in the transformed domain, which can be interpreted as finding a sparse representation with respect to the bases supported by singular vectors. This work aims to advance deep transform-based TNN methods by identifying a more compact representation through learnable bases, ultimately improving recovery accuracy. We specifically employ convolutional kernels as these learnable bases, demonstrating their capability to generate more compact representation, i.e., sparser coefficients of real-world tensor data compared to singular vectors. Our proposed model consists of two key components: a transform component, implemented through fully connected neural networks (FCNs), and a convolutional component that replaces traditional singular matrices. Then, this model is optimized using the ADAM algorithm directly on the incomplete tensor in a zero-shot manner, meaning all learnable parameters within the FCNs and convolution kernels are inferred solely from the observed data. Experimental results indicate that our method, with this straightforward yet effective modification, outperforms state-of-the-art approaches on video and multispectral image recovery tasks.

1. Introduction

In multidimensional data applications, tensors – higher-order generalizations of matrices – serve as fundamental data structures. These applications span a wide range of fields, including color image and video processing [1–4], hyperspectral data recovery and fusion [5–7], personalized web search [8,9], high-order web link analysis [10], magnetic resonance imaging (MRI) data recovery [11], and seismic data reconstruction [12], *etc.* Due to inherent constraints such as the acquisition conditions and transmission capacity limitations, multidimensional data frequently exhibit incompleteness or significant corruption. The challenges associated with these issues, as highlighted in [3], drive the need for advanced methods in this field. Effectively characterizing and harnessing the internal structure of complex multidimensional data is thus crucial for achieving accurate and reliable results.

Low-rank models have demonstrated their effectiveness in processing two-dimensional data from diverse sources [13,14]. Extending this concept from matrices to tensors allows for the integration of multi-linear structural information, making tensors a powerful tool for handling data that is both multi-model and multi-relational [15]. The idea of using tensor completion for image compression was proposed in [16], and the super-pixel approach was also later studied in [17]. However, extending the concept of low-rankness from matrices to tensors directly is not trivial, as defining the rank of a tensor uniquely remains a challenging problem. In recent decades, the most commonly used rank definitions are the CANDECOMP/PARAFAC (CP) rank [18,19] and the Tucker rank [20] (also referred to as "n-rank" [21]). The CP rank is determined by CP decomposition, but calculating the CP rank for a specific tensor is recognized as an NP-hard problem [22]. Conversely, Tucker rank is derived via Tucker decomposition, which requires unfolding the tensor along each mode, often disrupting its inherent structures. In this paper, we focus on an innovative alternative form of tensor rank, derived from the tensor singular value decomposition (t-SVD). Originally introduced by Braman et al. [23] and Kilmer et al. [24], the t-SVD relies on the tensor–tensor product (referred to as t-prod), which enables the integral manipulation of third-order tensors, thereby preserving the information that is often lost when tensors are converted into matrices or flattened [25]. Recent advancements have integrated the t-SVD with deep learning techniques to optimize the sparsity of singular values in recovered tensors by exploiting spatial dimensions more effectively [26]. Additionally, to

* Corresponding author.
E-mail addresses: lixizhuo0808@163.com (X.-Z. Li), taixiangjiang@gmail.com (T.-X. Jiang), liqiaoyoung@163.com (L. Yang), gliu@swufe.edu.cn (G. Liu).

https://doi.org/10.1016/j.neucom.2024.129036
Received 29 May 2024; Received in revised form 8 October 2024; Accepted 22 November 2024
Available online 29 November 2024
0925-2312/© 2024 Elsevier B.V. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

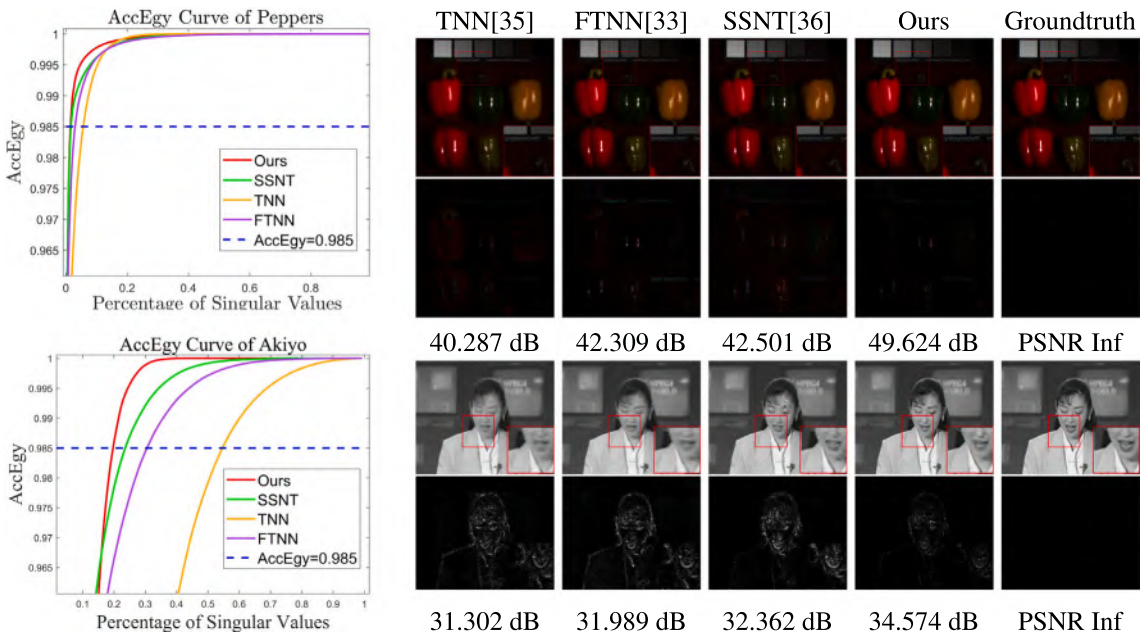

Fig. 1. The accumulation energy ratio (AccEgy) is calculated by considering the proportion of singular values derived from different transformations applied to the transformed frontal slices of the MSI *Peppers* (top left) and video *Foreman* (bottom left). The right side displays the restoration outcomes using several techniques on these two datasets with a sampling rate (SR) of 10%. The AccEgy of the k largest singular values is defined as $\sum_{i=1}^{k}\sigma_i^2/\sum_j\sigma_j^2$, where σ_i represents the ith singular value. In our method, σ_i refers to the ith representation coefficient after convolution. It can be observed that SSCNT exhibits a higher energy concentration, indicating that our method learns a more compact representation, leading to superior completion results.

enhance the robustness of tensor transformations, Saragadam et al. [27] proposed a novel framework that employs deep generative networks for low-rank decomposition of both matrices and tensors, demonstrating improved stability and decomposition quality. Furthermore, t-SVD has demonstrated an exceptional ability to capture the spatial shift correlations frequently present in real-world data [23,24,28].

In [29], Kernfeld et al. emphasized that the t-prod relies on a convolution-like operation, which can be efficiently executed by implementing the Discrete Fourier Transform (DFT). A practical approach for computing the t-prod of two tensors involves three key steps: (i) applying the Fast Fourier Transform (FFT) along the tubes of the tensors, (ii) performing matrix multiplications on each pair of frontal slices in the transform domain, and (iii) applying an inverse FFT along the tubes of the resulting tensor. Importantly, beyond the Fourier transform, the t-prod and the t-SVD framework can be established using any invertible linear transformation [29], such as the Discrete Cosine Transform (DCT). In [30], it was demonstrated that the mirror boundary conditions used in DCT are more effective in preserving the head and tail frontal slices compared to the periodic boundary conditions of DFT, resulting in improved performance in tensor completion. Additionally, studies such as [31,32] have derived robust theoretical guarantees by utilizing transformations like unitary and invertible linear transformations. However, the requirement of invertibility limits exploration of results obtained from non-invertible (or semi-invertible) transformations, potentially leading to redundant outcomes. To address this, Jiang et al. [33] proposed using the tight wavelet frame, also known as a framelet, as a transformation method, which effectively reduces redundancy in transformations. Furthermore, in their subsequent work [34], they explored a more efficient low-rank representation of the transformed tensor by employing a dictionary-based technique to obtain low-rank coding coefficients.

However, the linear structure of the aforementioned transformations limits their ability to accurately capture the intricate and nonlinear characteristics inherent in real-world data. To address this limitation, Luo et al. [35] introduced a nonlinear multilayer neural network approach, which significantly enhances the model's ability to represent the complexity and nonlinear features of real-world data. We begin by defining the forward transform f of the nonlinear transformation and its corresponding backward transform g. The model can then be formulated as follows:

$$\min_{f,g}\ \lambda\sum_{k=1}^{n_3}\|f(\mathcal{X})^{(k)}\|_* + L(g(f(\mathcal{X})),\mathcal{O}), \tag{1}$$

where $\mathcal{X}\in\mathbb{R}^{n_1\times n_2\times n_3}$ and $f(\mathcal{X})$ is the transformed tensor acquired through nonlinear multi-layer neural network. The superscript denotes the kth frontal slice of the transformed tensor. The term $L(g(f(\mathcal{X})),\mathcal{O})$ represents the fidelity loss function, and λ is the trade-off parameter. This model's multi-layer adaptive transformation enables it to achieve excellent results across a wide range of datasets and tasks. According to the definition of t-SVD, the kth frontal slice of the transformed tensor can be expressed as $f(\mathcal{X})^{(k)}=\mathbf{U}^{(k)}\mathbf{S}^{(k)}\mathbf{V}^{(k)H}$. Since minimizing $\|f(\mathcal{X})^{(k)}\|_*$ is equivalent to minimizing $\|\mathbf{S}\|_1$, the first term in model (1) is equivalent to $\lambda\sum_{k=1}^{n_3}\|\mathbf{U}^{(k)H}f(\mathcal{X})^{(k)}\mathbf{V}^{(k)}\|_1$. The unitary matrices $\mathbf{U}^{(k)}$ and $\mathbf{V}^{(k)}$ obtained through t-SVD are derived directly from the data, rather than being based on predefined paradigms. It is noteworthy that $\mathbf{U}^{(k)}$ and $\mathbf{V}^{(k)}$ can be respectively interpreted as row and column operations. The low-rankness of $f(\mathcal{X})^{(k)}$ is directly related to the sparsity of $\mathbf{S}$. Consequently, the objective of (1) is to find a sparse representation with respect to singular vectors in $\mathbf{U}^{(k)}$ and $\mathbf{V}^{(k)}$.

Considering the superiority of representation learning in processing high-dimensional data by reducing dimensionality and extracting significant features, several methodologies have been proposed to enhance data representation efficiency. For instance, a hierarchical subnet neural network based on progressive learning was introduced to optimize data representation [36]. Zhang et al. [37] further achieved a robust representation by simultaneously considering low-dimensional features and classifier models. Moreover, a technique utilizing subnets to construct discriminative latent spaces was proposed to address the challenges of loosely connected feature encoding [38]. In this work, we consider to find a more compact representation by introducing the convolutional kernels to replace the singular vectors. Specifically, our motivations involve three key aspects. First, the frontal slices of many real-world tensors, within the transform-based t-SVD framework, are not strictly low-rank. Second, minimizing the ℓ_1 norm of singular

values in the transform domain (i.e., minimizing the transform-based TNN) can indiscriminately shrink small singular values that are associated with intricate geometric features. Third, the convolution operation effectively replaces row and column operations and has been shown to excel in extracting spatial features. Considering these factors, we adopt convolution in place of $\mathbf{U}^{(k)H}$ and $\mathbf{V}^{(k)}$ within the transform-based t-SVD framework. To start, we employ a fully connected neural network (FCN) as the nonlinear transform, applied to both the forward and backward transformations. In the transformed domain, convolution operations are then used to replace the singular bases. Convolution is adaptive to data and is believed to effectively extract spatial features while preserving intricate textures. Thus, our approach seeks to find a more compact representation in the (nonlinear) transform domain by minimizing the representation coefficients with respect to the convolutional kernel. We refer to this transformation as the Self-Supervised Convolutional Nonlinear Transform (SSCNT). All parameters in the FCN and convolutional kernels can be inferred solely from the incomplete observation in a self-supervised (or zero-shot learning) manner. As illustrated in Fig. 1, which displays the accumulation energy (AccEgy) ratio of singular values from previous transform-based TNN methods and the AccEgy ratio of our representation coefficients, our method provides a more compact representation of videos and multispectral images (MSI), evidenced by higher AccEgy values with fewer proportional singular values (representation coefficients). This indicates that SSCNT is capable of identifying a more compact representation, leading to superior tensor completion results. The example shown in Fig. 1 demonstrates that our method, with a simple modification, significantly outperforms the compared methods in terms of peak signal-to-noise ratio (PSNR). This validates that finding a more compact representation is indeed beneficial for characterizing the inner structure of real-world tensor data.

We summarize the contributions of this paper as follows:

- We present a self-supervised, nonlinear transform-based TNN designed to effectively capture the intricate and nonlinear features of multi-dimensional images. This framework leverages a nonlinear multilayer neural network combined with a convolutional network to represent the transform. Crucially, the transform is learned directly from the observed data through self-supervision, eliminating the need for labeled datasets.
- Our approach enhances model capacity by generating more compact representations, achieved by minimizing the ℓ_1 norm of singular values in the transform domain. This technique replaces the traditional singular value decomposition with convolutional operations, streamlining the process while retaining performance. Extensive experiments across various multi-dimensional inverse problems demonstrate the clear superiority of our method compared to existing state-of-the-art techniques.

The structure of this paper is presented as follows. In Section 2, an introduction to tensors is provided, covering some basic background information. Section 3 introduces the proposed method. Section 4 contains the details of the experimental results. Lastly, concluding remarks are provided in Section 5.

2. Notations and preliminaries

This section outlines the fundamental elements necessary to implement the proposed method. We start by introducing the basic tensor notations, followed by an introduction to the t-SVD framework, which was originally proposed in the Refs. [24,25,39,40]. We are reiterating these concepts here for the convenience of the readers.

Floral letters, such as $\mathcal{X}$, are employed to represent tensors, while capital boldface letters, like $\mathbf{X}$, are used for matrices. The *i, j, k*th element of a third-order tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is denoted by $\mathcal{X}(i,j,k)$. The *i*th frontal slice of $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is denoted by $\mathcal{X}^{(i)} \in \mathbb{R}^{n_1 \times n_2}$.

Definition 1. Tensor–tensor Product (t-prod) [25]: The tensor–tensor product of $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{B} \in \mathbb{R}^{n_2 \times n_4 \times n_3}$, denoted as $\mathcal{C} = \mathcal{A} * \mathcal{B} \in \mathbb{R}^{n_1 \times n_4 \times n_3}$. The (i, j)th tube $\mathbf{c}_{ij:}$ is defined as follows:

$$\mathbf{c}_{ij:} = \mathcal{C}(i,j,:) = \sum_{k=1}^{n_2} \mathcal{A}(i,k,:) * \mathcal{B}(k,j,:), \tag{2}$$

where $*$ denotes the circular convolution between two same-size tubes.

Since the convolution operation is equivalent to an element-wise product in the Fourier domain, the t-prod between two tensors can be efficiently computed by performing matrix multiplication on their frontal slices after applying the DFT or FFT along the third mode.

Definition 2 (*Conjugate Transpose [25]*)**.** The conjugate transpose of $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, indicated as $\mathcal{A}^H$, is defined as follows: $(\mathcal{A}^H)^{(1)} = (\mathcal{A}^{(1)})^H$ and $(\mathcal{A}^H)^{(i)} = (\mathcal{A}^{(n_3+2-i)})^H (i = 2, \ldots, n_3)$.

Definition 3 (*Identity Tensor [25]*)**.** The identity tensor $\mathcal{I} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$ is a tensor that has the first frontal slice consisting of a $n_1 \times n_1$ identity matrix, while all other slices are filled with zeros.

Definition 4 (*Orthogonal Tensor [25]*)**.** The tensor $\mathcal{Q}$ is orthogonal if $\mathcal{Q} * \mathcal{Q}^H = \mathcal{Q}^H * \mathcal{Q} = \mathcal{I}$.

Definition 5 (*F-diagonal Tensor [25]*)**.** The tensor $\mathcal{A}$ defined as f-diagonal is characterized by the fact that every frontal slice $\mathcal{A}(i)$ is a diagonal matrix.

Theorem 1 (*T-SVD [24,25]*)**.** *The tensor singular value decomposition (t-SVD) of a tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is defined as follows:*

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

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

Definition 6 (*Tensor Tubal-rank [39]*)**.** The t-SVD of a tensor $\mathcal{A}$ is given by $\mathcal{U} * \mathcal{S} * \mathcal{V}^H$, where $\mathcal{A}$ is a tensor of size $n_1 \times n_2 \times n_3$. The tubal-rank of $\mathcal{A}$, denoted as $rank_t(\mathcal{A})$, is defined as the number of singular tubes in the tensor $\mathcal{S}$ that are not equal to zero.

Definition 7 (*Tensor Tubal Nuclear Norm (TNN) [41]*)**.** The tensor nuclear norm of a tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, represented by $\|\mathcal{A}\|_{\text{TNN}}$, is defined as

$$\|\mathcal{A}\|_{\text{TNN}} = \sum_{i=1}^{n_3} \|\tilde{\mathcal{X}}^{(i)}\|_*, \tag{4}$$

where $\tilde{\mathcal{X}}^{(i)}$ refers to the Fourier transformed tensor's frontal slice.

As noted by Kernfeld et al.[29], the implementation of the tensor–tensor product is not limited to the discrete Fourier transform, therefore the tensor nuclear norm can also be defined in different transformed domain [30,31,33–35].

3. Proposed method

This section presents the framework of the suggested nonlinear transform. By employing the developed transformation, we establish an optimization model for restoring low-rank tensors.

3.1. Revisiting TNN and SSNT

Zhang et al. [39] developed a convex replacement of the tensor tubal-rank, which they refer to as TNN. The formulation of the model is as follows:

$$\min_{\mathcal{X}} \lambda \|\mathcal{X}\|_{\text{TNN}}, \quad \text{s.t. } \mathcal{X}_\Omega = \mathcal{O}_\Omega, \tag{5}$$

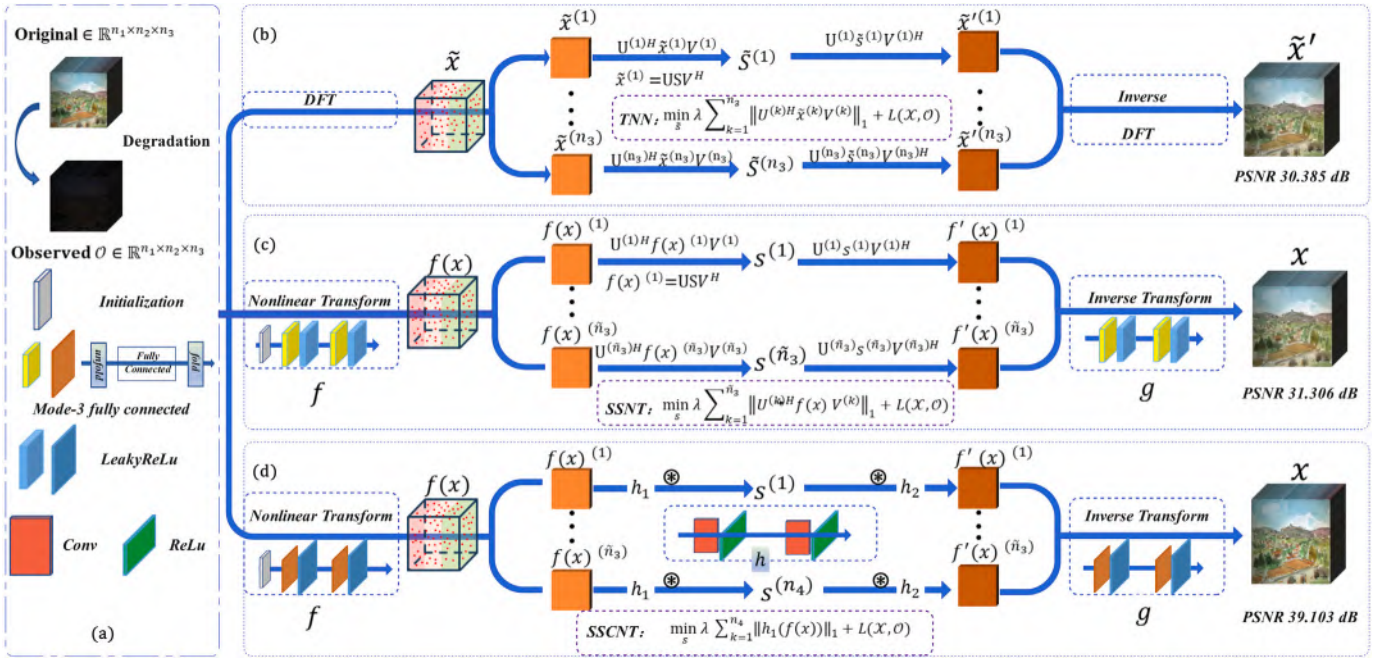

Fig. 2. The procedure for recovering low-rank tensors using linear transform-based TNN [39], SSNT [35], and the proposed SSCNT.(a) Degradation Process: The process where the sampling rate (SR) is 10%. (b) TNN Method: The TNN represents a standard linear transform-based approach for recovering low-rank tensors. (c) SSNT Method: The Self-Supervised Nonlinear Transform (SSNT) approach. (d) SSCNT Method: The proposed Self-Supervised Convolutional Nonlinear Transform (SSCNT) method.

where $\mathcal{O}$ represents the incomplete observations and Ω represents the observed set. Considering that the nuclear norm of a tensor is defined as the sum of the singular values of its frontal slices in the transformed domain, and given that t-SVD can be viewed as operating after a linear projection, it becomes necessary to employ alternative methods to capture the nonlinear relationships between the first and second dimensions of the tensor. For any tensor $\mathcal{X}$, its nuclear norm can be expressed as $\sum_k \mathbf{S}^{(k)} = \mathbf{U}^{(k)H}\mathcal{X}^{(k)}\mathbf{V}^{(k)}$. Therefore, model (5) can be restated in the following manner:

$$\min_{\substack{\mathcal{X} \\ \mathbf{U}^{(k)H}\mathbf{U}^{(k)}=\mathbf{I},\mathbf{V}^{(k)H}\mathbf{V}^{(k)}=\mathbf{I}}} \lambda \sum_{k=1}^{n_3} \|\mathbf{U}^{(k)H}\tilde{\mathcal{X}}^{(k)}\mathbf{V}^{(k)}\|_1, \quad \text{s.t. } \mathcal{X}_\Omega = \mathcal{O}_\Omega. \tag{6}$$

Luo et al. [35] proposed using the fully connected layer of nonlinear mode-3 ($NoFC_3$) being the unit of Self-Supervised Nonlinear Transform (SSNT). A single $NoFC_3$ layer can be represented as

$$w_i(\mathcal{X}) = \sigma_1(\mathcal{X} \times_3 \mathbf{W}_i), \tag{7}$$

where $\times_3$ indicates the mode-3 tensor-matrix product e.g., $\mathcal{X} \times_3 \mathbf{A} = \text{fo1d}_3(\mathbf{A} \times \text{unfo1d}_3(\mathcal{X}))$, where $\text{fold}_3(\cdot)$ is the mode-3 folding operator that transforms a matrix of size $\mathbb{R}^{n_3\times n_1 n_2}$ into a tensor of size $\mathbb{R}^{n_1\times n_2\times n_3}$. Similarly, the mode-3 unfolding operator $\text{unfold}_3(\cdot)$ is used to transform a tensor of size $\mathbb{R}^{n_1\times n_2\times n_3}$ into a matrix of size $\mathbb{R}^{n_3\times n_1 n_2}$. σ_1 represents the nonlinear activation function LeakyReLU [42], while $\mathbf{W}_i$ is the trainable matrix in the linear weighting operations layer.

Luo et al. [35] have constructed the SSNT by stacking two layers of $NoFC_3$ to represent the subspace accurately. The SSNT is a mapping function $f : \mathbb{R}^{n_1\times n_2\times n_3} \to \mathbb{R}^{n_1\times n_2\times \tilde{n}_3}$, which can be represented as

$$f(\mathcal{X}) = w_2 \circ w_1(\mathcal{X}), \tag{8}$$

where $\circ$ represents the composition of functions. We can use Θ_f to denote the learnable parameters in f. Similarly, the inverse transform $g : \mathbb{R}^{n_1\times n_2\times \tilde{n}_3} \to \mathbb{R}^{n_1\times n_2\times n_3}$ can be represented as

$$g(\mathcal{X}) = w_4 \circ w_3(\mathcal{X}), \tag{9}$$

with its learnable parameters denoted as Θ_g. Based on above, for the observed data $\mathcal{O} \in \mathbb{R}^{n_1\times n_2\times n_3}$, the SSNT optimization model for recovering low-rank tensors can be formulated as

$$\min_{\substack{\Theta_f,\Theta_g \\ \mathbf{U}^{(K)H}\mathbf{U}^{(K)}=\mathbf{I},\mathbf{V}^{(K)H}\mathbf{V}^{(K)}=\mathbf{I}}} \lambda \sum_{k=1}^{\tilde{n}_3} \|\mathbf{U}^{(k)H} f(\mathcal{X})\mathbf{V}^{(k)}\|_1 + L(g(f(\mathcal{X})),\mathcal{O}), \tag{10}$$

where $L(\mathcal{X},\mathcal{O})$ is the fidelity term, measuring the distance between the output after degradation and the observation.

3.2. SSCNT for low-rank tensor recovery

To improve the precision of nonlinear transformations, we propose a novel method called SSCNT, which leverages a combination of multi-layer neural networks and convolutional networks for transformation creation. As illustrated in Fig. 2, the SSCNT approach is compared against traditional linear transform-based TNN and SSNT for low-rank tensor recovery. This comparative analysis demonstrates the effectiveness of the proposed SSCNT in enhancing the accuracy of low-rank tensor reconstruction.

In this context, we replace $\mathbf{U}^{(k)H}$ and $\mathbf{V}^{(k)}$ in model (10) with convolution operations. The proposed SSCNT features a hierarchical structure that includes linear weights, nonlinear transformations, and nonlinear activation functions. Specifically, we introduce the linear weighting operation (*ConR*) layer as a crucial component of SSCNT. A single *ConR* layer is formulated as follows:

$$v_i(\mathcal{X}) = \sigma_2(\mathcal{X} \circledast \mathbf{k}),\ i = 1,2,3,4, \tag{11}$$

where σ_2 is used to represent the nonlinear activation function ReLu, $\mathbf{k}$ stands for the convolution kernel, and $\circledast$ is used to represents the 2-D spatial convolution operation. Our approach aligns with traditional TNN methodologies by employing a neural network on mode-3 to analyze interactions among frontal slices.

To capture the nonlinear relationships between the first and second dimensions, we stack two *ConR* layers to construct the proposed SSCNT $h_1 : \mathbb{R}^{n_1\times n_2\times \tilde{n}_3} \to \mathbb{R}^{n_1\times n_2\times n_4}$, which is designed as

$$h_1(\mathcal{X}) = v_2 \circ v_1(\mathcal{X}), \tag{12}$$

with Θ_{h_1} denoting learnable parameters in h_1. Similarly, we stack two *ConR* layers to construct the inverse transform $h_2 : \mathbb{R}^{n_1\times n_2\times n_4} \to \mathbb{R}^{n_1\times n_2\times \tilde{n}_3}$, which is designed as

$$h_2(\mathcal{X}) = v_4 \circ v_3(\mathcal{X}), \tag{13}$$

with Θ_{h_2} denoting the learnable parameters in h_2.

Furthermore, although the data generally exhibits low-dimensional characteristics, missing observations necessitate transforming the data into a lower-dimensional space to effectively utilize its structure. When the properties of the lower-dimensional space are accurately captured, the reconstructed image will demonstrate improved filling effects after inverse projection. However, subspace characterization faces two primary challenges: first, the incomplete data complicates the identification of an appropriate lower-dimensional space; second, the subspace obtained via t-SVD transformation may lack sufficient compactness. Moreover, the unitary matrices $\mathbf{U}^{(k)}$ and $\mathbf{V}^{(k)}$, as row and column operations, are unnecessary and restrictive in promoting sparsity in model (10). Therefore, by substituting the linear projection with nonlinear convolution, convolution is employed as the projection operator to replace $\mathbf{U}^{(k)}$ and $\mathbf{V}^{(k)}$.

As discussed above, given the observed data $\mathcal{O} \in \mathbb{R}^{n_1\times n_2\times n_3}$, the proposed framework for recovering low-rank tensors is defined by the following optimization problem

$$\min_{\Theta_f,\Theta_g,\Theta_{h_1},\Theta_{h_2},\mathcal{X}} \lambda \sum_{k=1}^{n_4} \|h_1(f(\mathcal{X}))\|_1 + L(\Phi(\mathcal{X}),\mathcal{O}), \tag{14}$$

where Φ represents the sequential composition of g, h_2, h_1, and f, i.e., $\Phi(\mathcal{X}) = g\left(h_2\left(h_1\left(f\left(\mathcal{X}\right)\right)\right)\right)$.

In this optimization problem, the fidelity term for tensor completion is formulated as

$$L(\mathcal{X},\mathcal{O}) = \|\mathcal{W}\odot(\mathcal{X}-\mathcal{O})\|_F^2, \tag{15}$$

where $\mathcal{W} \in \mathbb{R}^{n_1\times n_2\times n_3}$ is a 0–1 weighting tensor, with entries set to 1 for observed elements and 0 for unobserved elements.

We use the adaptive moment estimation (Adam) [43] to optimize (14) with all parameters in f, h_1, h_2, and g randomly initialized. To obtain a better initial guess of $\mathcal{X}$, we use the simple linear interpolation introduced in [34]. The pseudocode is summarized in Algorithm 1.

To assess the efficiency of our algorithm, we analyze its computational complexity. Given a 3rd-order tensor $\mathcal{X} \in \mathbb{R}^{n_1\times n_2\times n_3}$, the third dimension is first transformed by the function f, resulting in $\tilde{n}_3$, and then further transformed by the function g to produce n_4. The time and space complexities in each iteration can be broken down into following components:

- For the functions f and g that contain fully connected layers, the time complexity is $O(n_1n_2n_3)$ and $O(n_1n_2\tilde{n}_3)$, respectively, while the space complexity is $O(n_1n_2 + n_1n_3 + n_2n_3)$ and $O(n_1n_2 + n_1\tilde{n}_3 + n_2\tilde{n}_3)$, respectively.
- For the functions h_1 and h_2 that contain convolutional layers, both the time complexity and space complexity are $O(n_1n_2\tilde{n}_3n_4)$ and $O(n_1n_2(\tilde{n}_3 + n_4))$, respectively.

Overall, the total time complexity of our algorithm is $O(n_1n_2(n_3 + \tilde{n}_3 + \tilde{n}_3n_4))$, and the total space complexity is $O(n_1n_2(\tilde{n}_3+n_4+1)+(n_1+n_2)(n_3+\tilde{n}_3))$.

The proposed approach offers a more compact representation compared to existing transform-based TNN methods. Insights drawn from Fig. 1 (left side) demonstrate that our SSCNT method results in a higher concentration of energy within the transformed tensor, as opposed to conventional linear transform-based TNN techniques. This concentration suggests that our method achieves a sparser representation. Specifically, as indicated by the auxiliary dashed lines, our method requires a smaller proportion of singular values to capture 98.5% of the total energy in the data, compared to other methods. Consequently, our approach is expected to deliver a more accurate tensor low tubal-rank approximation while offering a more flexible representation. This significant improvement is a key contribution and serves as the driving force behind our proposed method.

Algorithm 1 SSCNT for Tensor Completion

Input: The observed tensor $\mathcal{O}$ with the support of observed entries Ω; trade-off parameters λ; maximum iteration number $t_{\max}$.
Initialization: $\mathcal{X}^0 = \text{Init}(\mathcal{O})$ ▷ Linear interpolation $\Theta_f^0, \Theta_g^0, \Theta_{h_1}^0$ and $\Theta_{h_2}^0$ are randomly initialized. ▷ Kaiming initialization
$t = 0$;
while $t < t_{\max}$ **do**
    $t = t + 1$;
    Update $\mathcal{X}^t, \Theta_f^t, \Theta_g^t, \Theta_{h_1}^t$ and $\Theta_{h_2}^t$ by minimizing (14) via Adam;
end while
Output: The recovered tensor $\Phi(\mathcal{X}^t)$, where Φ represent the composition of g, h_2, h_1, and f, with the parameters Θ_g^t, $\Theta_{h_2}^t$, $\Theta_{h_1}^t$, and Θ_f^t, respectively .

3.3. Implementation

We emphasize that the proposed SSCNT is trained exclusively using the observed data in a fully self-supervised manner. As a result, there is no need for additional training data or the conventional division of data into training and testing sets. A critical aspect of our network's structure is the integration of both linear and convolutional layers. For the linear layer, we employ the efficient strategy given in [35] . In specific, each $NoFC_3$ is composed of a fully connected network,[1] whereas each proposed $ConR$ is composed of a convolution block.[2] Further, let us consider $\mathcal{X} \in \mathbb{R}^{n_1\times n_2\times \tilde{n}_3}$ as an example to demonstrate the convolutional layer. In order to leverage the inherent nonlinear characteristics of real data, we design a nonlinear mapping to the transformed data. The 2-D spatial convolution layer is executed by convolving each individual frontal slice $\mathcal{X}^{(i)}$ with various $3 \times 3 \times 1$ filters (in total $\tilde{n}_3$ filters). This process produces a feature map of size $n_1 \times n_2 \times n_4$, which effectively captures the local spatial correlation of the data. In the proposed method, we set $\lambda = N \times 10^{-7}$, where $N = n_1 \times n_2 \times n_3$, $\tilde{n}3 = 2n_3$, and $n_4 = 90$ and 250 for the CAVE and video datasets, respectively. To optimize the process, we employ the Adam optimizer, with termination based on a stopping criterion defined as a maximum number of iterations, $t_{max} = 7000$. Given the non-convex nature of model (14), the initialization of $\mathcal{X}_0$ in Eq. (8) holds significant importance. Hence, $\mathcal{X}_0$ is set to the result of the initialization function Init($\mathcal{O}$). For tensor completion, the initialization function Init($\cdot$) refers to the linear interpolation method given in [34]. This method offers an optimal initialization while also minimizing time consumption. All experiments were conducted on the platform of Windows 11 with an Intel(R) Core i5-9400f CPU, and RTX 3090 GPU with 64 GB RAM, and the quantitative metrics were computed on Matlab 2021b.

4. Experiments

4.1. Experimental settings

We employ two traditional datasets to evaluate the effectiveness of the suggested methods, namely the Columbia multispectral database (CAVE[3]) dataset and video dataset.[4] We evaluate all methods on different datasets using a range of sample rates (SRs): 5%, 10%, 15%,

[1] nn.Linear($\cdot$) in Pytorch.
[2] nn.Conv2d($\cdot$) in Pytorch.
[3] http://www.cs.columbia.edu/CAVE/databases/multispectral.
[4] http://trace.eas.asu.edu/yuv/.

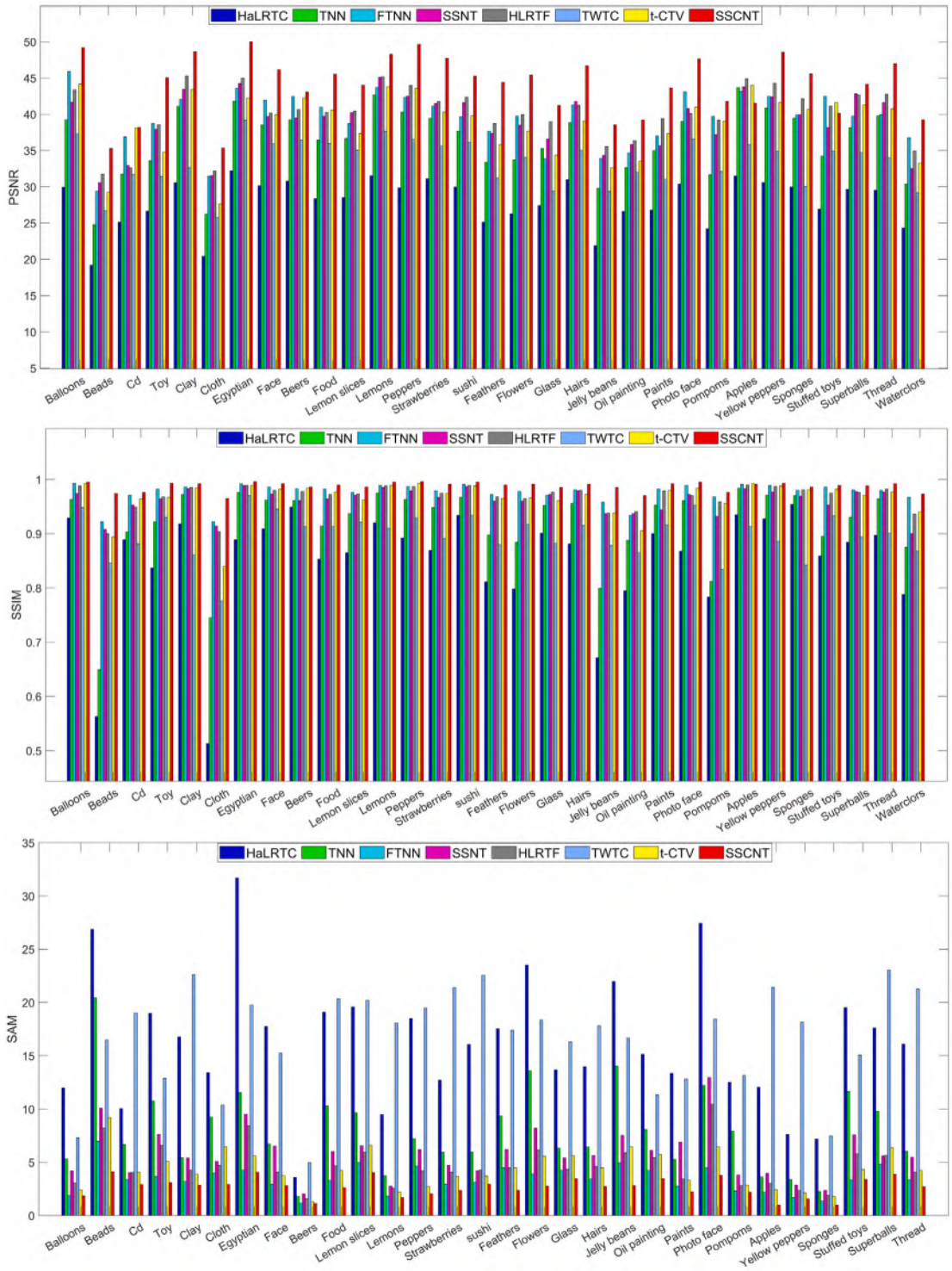

Fig. 3. The PSNR, SSIM, and SAM values representing the restored outcomes are derived from various techniques applied to the CAVE dataset with a 10% SR.

20%, and 25%. Before the experiment, the pixel values of all datasets have been normalized to a range of [0, 1] band-by-band. In addition, we employ three quantitative evaluation indicators: peak signal-to-noise ratio (PSNR), structural similarity (SSIM) [44], and spectral angle mapper (SAM) [45]. Higher PSNR and SSIM values indicate superior quality, while lower SAM values signify a narrower spectral angle between the original data and the restored outcome.

To assess the effectiveness of the proposed SSCNT, we compare it with seven recent and state-of-the-art TC methods. These methods include HaLRTC [46], a specialized algorithm for tensor filling, as well as various linear transform-based approaches. The following methods are referenced: TNN [47], TWTC [48] (a decomposition method that initiates with the structure of an analytical diagram), t-CTV [49] (a low-rank and smooth tensor recovery method), SSNT [35] (a Self-Supervised Nonlinear Transform-Based Tensor Nuclear Norm method), FTNN [33] (where the TNN induced by framelet transform), and HLRTF [50] (which embedded a DNN as a nonlinear transform into the t-SVD framework). The authors have either published the codes

Table 1
The PSNR, SSIM, and SAM values of the recovered results are obtained by different methods on MSI data. The red, blue, and green colors stand for the best, second best, and third best values, respectively.

Data	SR	0.05			0.1			0.15			0.2			0.25			Time
	Metric	PSNR	SSIM	SAM	PSNR	SSIM	SAM	PSNR	SSIM	SAM	PSNR	SSIM	SAM	PSNR	SSIM	SAM	(s)
Flowers	Observed	13.473	0.381	–	13.976	0.392	–	14.224	0.421	–	14.487	0.448	–	14.767	0.473	–	–
	HaLRTC [46]	20.863	0.672	34.885	26.268	0.798	23.512	29.635	0.870	15.747	31.880	0.906	12.325	33.586	0.928	10.445	36
	TNN [39]	25.585	0.686	25.089	33.688	0.884	13.600	37.019	0.936	10.039	39.154	0.957	8.123	40.993	0.970	6.793	208
	FTNN [33]	34.708	0.942	5.905	39.746	0.978	3.907	43.256	0.988	3.065	45.740	0.993	2.550	47.724	0.995	2.217	432
	SSNT [35]	34.282	0.914	13.989	38.481	0.960	8.193	41.890	0.980	4.373	44.423	0.987	3.493	45.751	0.991	3.025	245
	HLRTF [50]	36.137	0.932	7.909	39.959	0.965	6.164	42.769	0.982	4.804	44.310	0.984	4.145	45.801	0.989	3.518	72
	TWTC [48]	31.082	0.866	23.218	34.068	0.917	18.355	36.001	0.938	15.554	37.452	0.951	13.351	38.701	0.959	11.687	–
	t-CTV [49]	33.234	0.927	7.638	37.679	0.966	5.565	40.643	0.980	4.522	42.814	0.987	3.858	44.592	0.990	3.391	313
	Ours	40.526	0.987	3.925	45.411	0.991	2.763	46.983	0.994	2.481	48.226	0.995	2.313	48.788	0.995	2.221	243
Beads	Observed	13.182	0.143	–	14.292	0.178	–	14.545	0.214	–	14.808	0.250	–	15.088	0.285	–	–
	HaLRTC [46]	16.575	0.451	33.509	19.204	0.563	26.854	21.944	0.691	20.341	24.241	0.783	15.849	26.217	0.846	12.765	24
	TNN [39]	19.744	0.402	31.019	24.775	0.649	20.447	28.232	0.786	14.944	30.407	0.855	11.963	32.021	0.893	10.125	218
	FTNN [33]	25.164	0.807	11.267	29.377	0.922	6.974	32.498	0.960	5.043	34.836	0.976	3.924	36.955	0.985	3.209	396
	SSNT [35]	26.099	0.782	16.724	30.573	0.908	10.069	33.644	0.947	6.808	35.563	0.964	5.288	37.424	0.975	4.171	238
	HLRTF [50]	26.497	0.722	15.409	31.770	0.900	8.189	34.806	0.950	5.772	36.597	0.965	4.480	37.740	0.973	4.338	72
	TWTC [48]	24.058	0.741	21.383	26.734	0.846	16.469	28.561	0.893	13.638	29.909	0.919	11.650	31.066	0.935	10.198	–
	t-CTV [49]	25.511	0.803	12.791	29.226	0.894	9.185	31.641	0.935	7.273	33.495	0.956	6.021	35.067	0.968	5.169	317
	Ours	31.428	0.942	5.814	35.277	0.974	4.114	37.520	0.984	3.432	38.903	0.988	3.090	40.542	0.992	2.429	242
Balloons	Observed	13.281	0.125	–	13.737	0.161	–	13.986	0.188	–	14.246	0.212	–	14.527	0.233	–	–
	HaLRTC [46]	24.870	0.869	16.494	29.952	0.929	11.985	32.492	0.951	9.398	34.370	0.963	7.730	35.868	0.970	6.590	36
	TNN [39]	30.374	0.846	11.350	39.249	0.963	5.308	42.618	0.980	3.803	45.014	0.987	3.032	46.808	0.991	2.575	190
	FTNN [33]	41.215	0.984	2.600	45.929	0.993	1.882	48.282	0.995	1.699	50.309	0.996	1.412	51.497	0.997	1.281	485
	SSNT [35]	37.119	0.938	7.922	41.643	0.974	4.193	43.910	0.983	3.285	46.119	0.989	2.577	47.332	0.991	2.419	235
	HLRTF [50]	40.322	0.981	4.018	43.396	0.988	3.051	46.518	0.992	2.625	47.819	0.994	2.249	48.710	0.994	2.171	72
	TWTC [48]	34.400	0.922	9.205	37.275	0.948	7.298	39.044	0.957	6.248	40.296	0.963	5.536	41.419	0.966	4.980	–
	t-CTV [49]	39.585	0.983	3.239	44.164	0.992	2.409	46.765	0.994	1.997	48.657	0.996	1.748	50.127	0.996	1.586	307
	Ours	46.233	0.992	2.242	49.163	0.995	1.832	50.257	0.996	1.679	50.533	0.996	1.565	51.869	0.997	1.445	242
Feathers	Observed	12.872	0.207	–	13.361	0.236	–	13.613	0.273	–	13.878	0.308	–	14.154	0.341	–	–
	HaLRTC [46]	20.013	0.674	30.737	25.150	0.811	17.526	29.440	0.878	12.058	30.876	0.915	9.290	32.682	0.936	7.750	32
	TNN [39]	24.908	0.700	18.897	33.385	0.898	9.353	36.424	0.942	6.911	38.399	0.960	5.611	40.129	0.971	4.739	206
	FTNN [33]	32.131	0.926	8.959	37.664	0.972	4.493	41.353	0.985	3.091	44.142	0.991	2.412	46.389	0.994	1.988	230
	SSNT [35]	32.237	0.906	11.834	37.331	0.960	6.193	40.391	0.977	4.138	42.734	0.984	3.347	44.489	0.988	2.695	243
	HLRTF [50]	34.759	0.937	5.833	38.721	0.968	4.486	41.798	0.983	3.300	43.587	0.988	2.815	44.699	0.990	2.590	72
	TWTC [48]	28.514	0.820	21.583	31.190	0.880	17.386	32.946	0.908	14.765	34.348	0.925	12.746	35.504	0.936	11.180	–
	t-CTV [49]	31.357	0.928	6.203	35.792	0.965	4.490	38.614	0.978	3.610	40.781	0.985	3.069	42.550	0.989	2.678	310
	Ours	40.069	0.978	3.163	44.385	0.990	2.367	46.576	0.994	2.045	47.805	0.995	1.845	48.594	0.996	1.757	242

of comparison methods or acquired them from the corresponding homepages.

4.2. MSIs dataset

In this section, we assess the effectiveness of the proposed method by comparing it with other tensor completion techniques on the CAVE dataset. The CAVE dataset contains 32 Multispectral Images (MSIs), each with an initial data size of $512 \times 512 \times 31$. The collection has a highly intricate structure and contains detailed texture information.

In Fig. 3, the PSNR, SSIM, and SAM values of the restored results obtained through various techniques on the MSIs with a SR of 10% are displayed. It is apparent that the proposed method (highlighted by the red columns) consistently attains the highest and second-highest values across all data points, confirming the superior performance of the proposed methods.

The numerical results of the completion of the MSIs are presented in Table 1. These results demonstrate that the proposed SSCNT outperforms other approaches in terms of PSNR and SSIM values, highlighting its superior accuracy in recovering low-rank tensors. Moreover, SSCNT achieves exceptional SAM values, further showcasing its ability to effectively leverage the correlation along the third mode. Fig. 4 displays visual illustrations of the outcomes for tensor completion. To enhance the visualization, two specific local areas are selected and magnified beneath each image. Due to page limits, in Fig. 4, we only present the visual results on two scenarios for images (*Beads* and *Watercolors*) from the CAVE dataset. It is evident that SSCNT outperforms other approaches in image recovery. The proposed method excels in spatial domain recovery, particularly as demonstrated by the results on *Beads*. This improvement is attributed to the nonlinear convolution, which provides a more accurate representation of low-dimensional space.

To evaluate the performance of our method under more challenging structural missing patterns, we considered two specific patterns: one involving oblique lines for image (*Balloons*) and another with a flower-like structure for image (*Flowers*). As illustrated in Fig. 5, our method successfully addresses these complex cases, demonstrating its robustness in scenarios with structural pixel removal.

4.3. Video dataset

Within this section, we choose three distinct videos to conduct additional validation of the proposed strategies, namely, "*Akiyo*", "*Carphone*", and "*Foreman*". The dimensions of all the data are $144 \times 176 \times 100$.

To conduct further validation of the proposed methods. Quantitative assessment results of different methods applied to video datasets under various SRs are displayed in Table 2. It is evident that SSCNT consistently delivers the highest and second-highest results in most cases. Meanwhile, to evaluate the efficacy of SSCNT on video datasets, the restoration outcomes of different approaches on the "*Carphone*" and "*Foreman*" are presented in Fig. 6. It is evident that SSCNT performs admirably in terms of spatial details when compared to other approaches. In addition, the proposed method enhances the clarity of spatial features and textures.

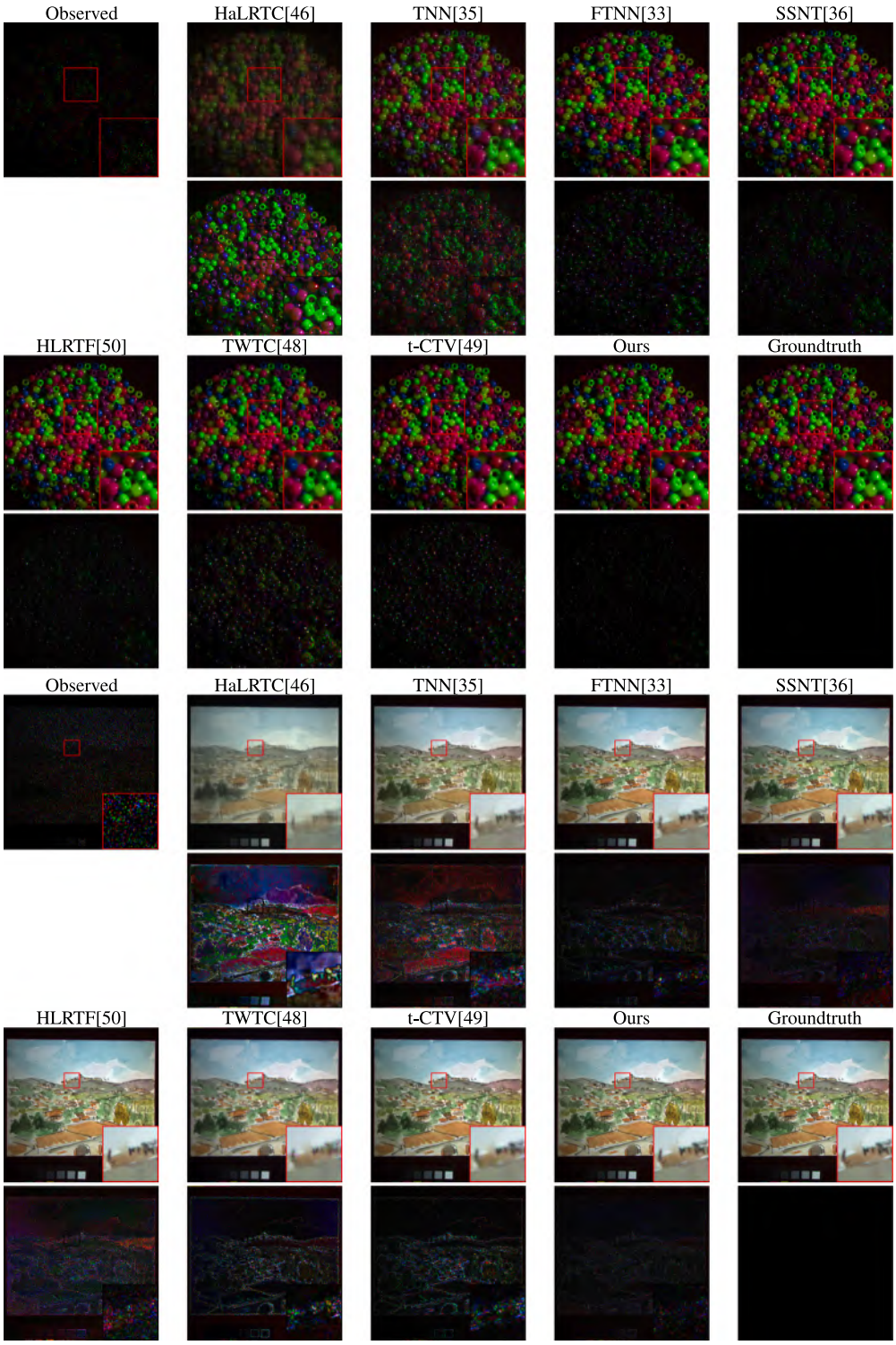

Fig. 4. Pseudo color images (composed of the 35th, 15th, and fifth bands) of recovered results and corresponding residual errors by different methods on MSI *Beads* and *Watercolors* with SR = 10%.

4.4. Discussions

To confirm the effectiveness of nonlinearity in the proposed SSCNT, ablation studies are carried out to evaluate the performance of several comparable solvers, including the proposed SSCNT, SSCNT with different nonlinear activation functions, SSCNT without any nonlinear activation functions and SSCNT without nonlinear transformations. Table 3 presents the quantitative evaluation outcomes of various techniques on MSI *Toy* with SR = 10%. Referring to Table 3, it is clear that the utilization of nonlinear activation functions leads to a substantial enhancement in overall performance, attributable to the strong modeling capabilities of nonlinearity. Additionally, using SSCNT with the ReLU activation function yields superior performance compared to other nonlinear activation functions such as LeakyReLU, PReLU, and ReLU6. Consequently, ReLU is selected as the activation function in SSCNT across all our experiments. Meanwhile, we can observe that SSCNT

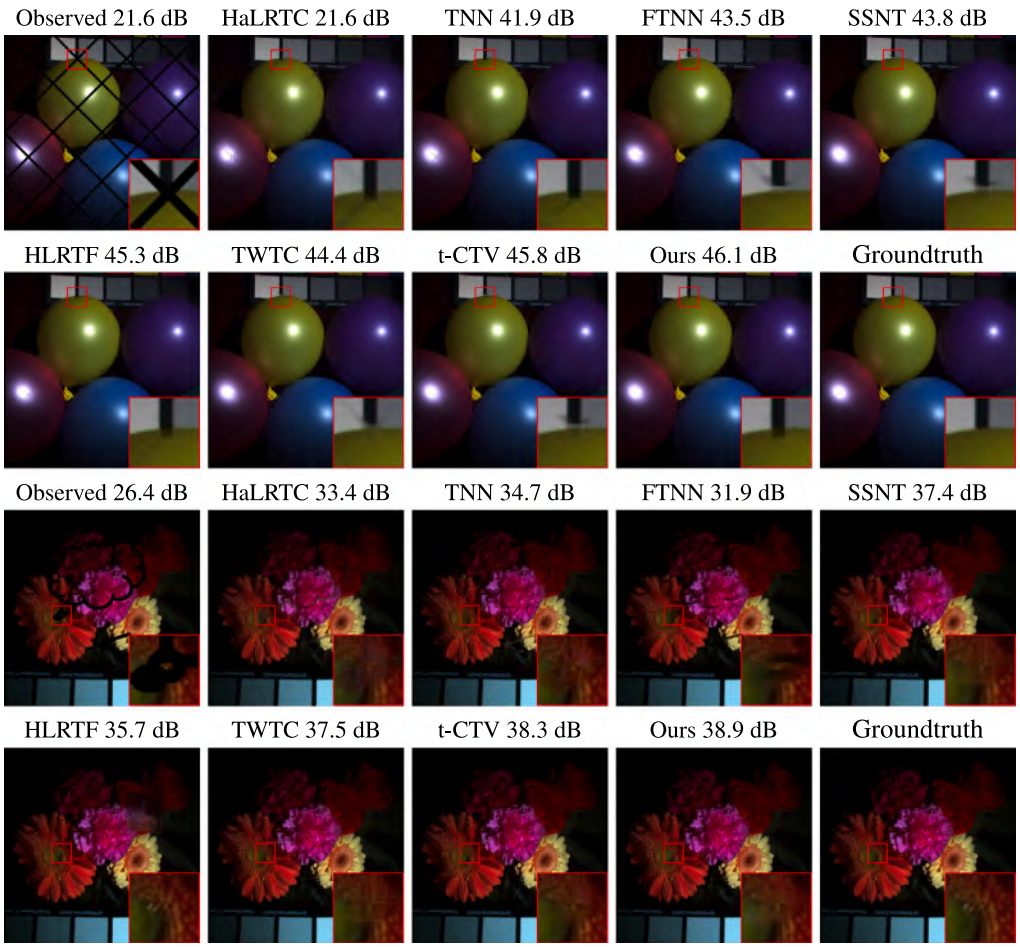

Fig. 5. The recovered results by different methods for tensor completion on *Balloons* and *Flowers* with structure missing patterns.

Table 2
The PSNR, SSIM, and SAM values of the recovered results are obtained by different methods on videos. The red, blue, and green colors stand for the best, second best, and third best values, respectively.

Data	SR	0.05		0.1		0.15		0.2		0.25		Time
	Method	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	(s)
Akiyo	Observed	6.609	0.012	6.845	0.019	7.091	0.025	7.359	0.031	7.632	0.038	–
	HaLRTC [46]	22.063	0.729	25.170	0.820	27.090	0.869	28.450	0.900	29.631	0.920	9
	TNN [39]	19.356	0.576	31.302	0.929	33.589	0.955	35.340	0.968	36.780	0.976	50
	FTNN [33]	29.026	0.904	31.989	0.952	34.326	0.971	36.353	0.981	38.004	0.987	262
	SSNT [35]	29.871	0.935	32.362	0.961	34.641	0.974	36.568	0.982	38.269	0.986	52
	HLRTF [50]	30.942	0.932	34.307	0.967	36.172	0.977	37.933	0.983	39.278	0.987	19
	TWTC [48]	26.994	0.848	28.850	0.904	30.234	0.931	31.376	0.948	32.317	0.959	–
	t-CTV [49]	30.505	0.929	30.429	0.932	32.934	0.958	35.039	0.973	36.713	0.980	75
	Ours	32.010	0.954	34.547	0.972	36.051	0.982	38.013	0.987	39.540	0.990	110
Carphone	Observed	6.398	0.013	6.633	0.022	6.878	0.029	7.140	0.037	7.424	0.045	–
	HaLRTC [46]	19.651	0.635	22.619	0.738	24.544	0.797	25.945	0.836	27.099	0.864	10
	TNN [39]	19.395	0.423	26.390	0.763	28.357	0.822	29.486	0.853	30.459	0.875	48
	FTNN [33]	26.458	0.826	28.936	0.890	30.637	0.919	31.964	0.928	33.220	0.940	291
	SSNT [35]	25.834	0.765	27.330	0.808	28.209	0.827	29.192	0.851	30.171	0.869	49
	HLRTF [50]	25.953	0.723	28.577	0.816	30.080	0.853	31.148	0.875	32.230	0.897	19
	TWTC [48]	25.500	0.809	27.364	0.871	28.630	0.902	29.600	0.921	30.465	0.934	–
	t-CTV [49]	25.174	0.810	28.516	0.880	30.395	0.911	31.635	0.917	32.724	0.929	77
	Ours	27.869	0.843	29.744	0.877	31.201	0.901	32.213	0.937	33.280	0.947	109
Foreman	Observed	3.813	0.006	4.049	0.009	4.299	0.013	4.560	0.017	4.840	0.021	–
	HaLRTC [46]	17.680	0.474	19.869	0.560	21.596	0.642	23.049	0.710	24.261	0.761	8
	TNN [39]	12.998	0.154	22.356	0.531	25.233	0.680	26.700	0.743	27.879	0.787	46
	FTNN [33]	23.916	0.692	26.460	0.796	28.270	0.857	29.819	0.892	31.155	0.917	270
	SSNT [35]	21.854	0.506	23.286	0.575	24.096	0.615	24.585	0.639	25.289	0.668	48
	HLRTF [50]	21.066	0.397	23.457	0.529	24.628	0.590	26.419	0.684	28.207	0.763	18
	TWTC [48]	23.673	0.720	25.623	0.815	26.983	0.862	28.082	0.890	29.025	0.910	–
	t-CTV [49]	23.377	0.718	26.355	0.809	28.056	0.853	29.343	0.881	30.538	0.904	75
	Ours	24.491	0.667	26.513	0.798	28.271	0.807	29.840	0.882	31.241	0.906	109

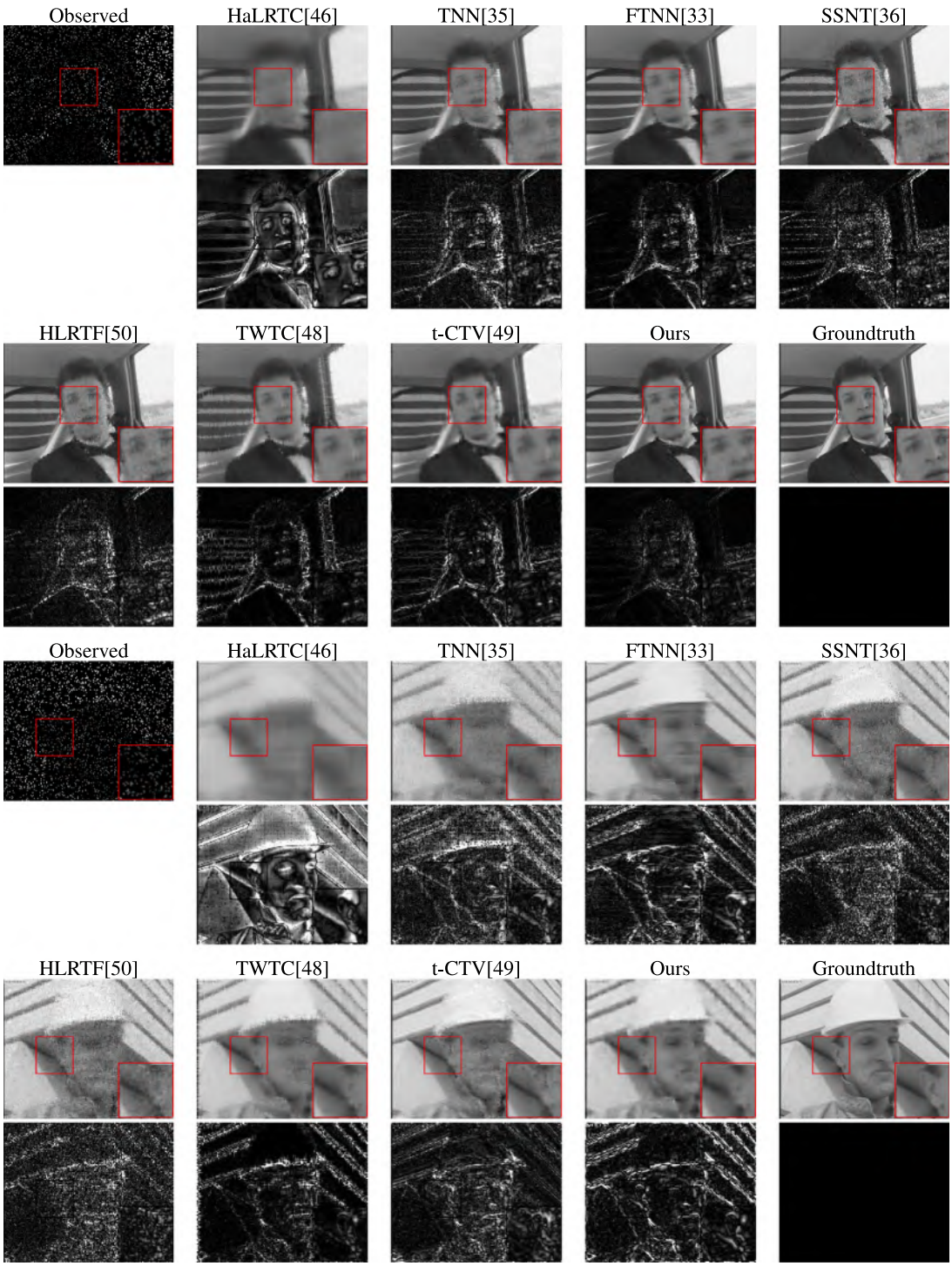

Fig. 6. Visualization of the recovery results and corresponding residual errors from various methods on video dataset with a 10% SR. From top to bottom: the 15th frame of *Carphone*, and 73th frame of *Foreman*, respectively.

(ReLu) considerably outperforms SSCNT (Linear), which reveals that nonlinear transformations is more effective to represent the third-order tensor in our framework.

Additionally, to explore the effectiveness of convolutions used to reconstruct images in the proposed method, we perform experiments to analyze the performance of convolutional kernels with varying sizes and the number of convolution layers. The outcomes are presented in the middle and lower section of Table 3. When the parameter p is at its lowest value, increasing p can improve the numerical results. However, as the value of p continues to increase, the numerical outcomes fail to meet our expectations. A potential explanation is that as the size of the convolution kernel increases, it becomes capable of capturing larger receptive fields, leading to a better understanding of the overall structure of the input data. However, when the convolution kernel becomes excessively large, it may overlook certain details and local characteristics, thereby diminishing the network's nonlinearity and its ability to express features. But the best result is achieved when the parameter q is at its lowest value. The possible reason is that as the number of convolutional layers increases, the complexity of the model also increases and after each convolutional layer, the feature map

Table 3
The quantitative results for tensor completion on MSI *Toy* with SR=0.1. The red, blue, and green colors stand for the best, second best, and third best values, respectively. The upper section of the table compares various activation functions employed in the convolutional layer, where *Linear* denotes the substitution of the f and g with a linear transformation, and *Null* indicates the SSCNT model without the functions f and g. The middle section examines the impact of different convolutional kernel sizes, where Conv(p) represents the size of the convolutional kernels used in the SSCNT reconstruction process. The lower section assesses the influence of the number of convolutional layers q, with $q = 0$ indicating SSCNT without convolutional components.

	Method	PSNR	SSIM	SAM	Time
Nonlinearity	Null	33.796	0.900	15.026	359
	ReLu	44.723	0.992	3.192	356
	LeakyReLu	43.529	0.991	3.631	363
	PReLu	42.527	0.988	3.822	452
	ReLu6	44.542	0.992	3.275	375
	Linear	42.701	0.989	4.275	236
SSCNT	Conv (1)	33.268	0.987	4.501	379
	Conv (2)	44.097	0.992	3.332	366
	Conv (3)	44.723	0.992	3.192	356
	Conv (4)	44.817	0.993	3.311	361
	Conv (5)	43.906	0.991	3.544	363
Hierarchy	$q = 0$	37.935	0.964	7.618	246
	$q = 1$	44.732	0.992	3.192	356
	$q = 2$	44.623	0.992	3.307	451
	$q = 3$	43.939	0.988	3.566	640
	$q = 4$	43.215	0.981	3.701	791
	$q = 5$	42.688	0.972	3.801	993

undergoes downsampling, which can result in some information loss. Excessive layers may lead to the loss of critical information, making it difficult for the network to capture important features.

5. Conclusion

This paper proposes a self-supervised TNN that utilizes convolutional nonlinear transformations. Specifically, a basis with a greater model capacity is utilized to generate a more condensed representation by substituting the singular value decomposition process with a convolution to minimize the ℓ_1 norm of singular values in the transform domain. In this way, the proposed method enables a more compact representation of real-world tensor data. Consequently, it proves to be highly effective in addressing tensor completion problems. Comprehensive experiments illustrate that the proposed method outperforms state-of-the-art methods in terms of performance.

CRediT authorship contribution statement

Xi-Zhuo Li: Writing – original draft, Methodology, Conceptualization. **Tai-Xiang Jiang:** Writing – review & editing, Visualization, Supervision, Methodology. **Liqiao Yang:** Writing – review & editing, Software, Methodology. **Guisong Liu:** Visualization, Supervision, Investigation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to thank the editor and reviewers for giving them many comments and suggestions, which are of great value for improving the quality of this manuscript. The authors wish to express their gratitude to the authors of [33,35,39,46,48–50] for providing their codes or data for this study. This work was supported in part by the Sichuan Science and Technology Program under Grant 2024ZYD0147, in part by the Natural Science Foundation of Xinjiang Uygur Autonomous Region under Grant 2024D01A18, in part by the National Natural Science Foundation of China (NSFC) under Grant 62376228, in part by Chengdu Science and Technology Program under Grant 2023 JB00-00016-GX, and in part by the Guanghua Talent Project.

Data availability

Data will be made available on request.

References

[1] A. Sobral, E.-h. Zahzah, Matrix and tensor completion algorithms for background model initialization: A comparative evaluation, Pattern Recognit. Lett. 96 (2017) 22–33.
[2] T. Korah, C. Rasmussen, Spatiotemporal inpainting for recovering texture maps of occluded building facades, IEEE Trans. Image Process. 16 (9) (2007) 2262–2271.
[3] J. Liu, P. Musialski, P. Wonka, J. Ye, Tensor completion for estimating missing values in visual data, IEEE Trans. Pattern Anal. Mach. Intell. 35 (1) (2012) 208–220.
[4] T.-X. Jiang, M.K. Ng, J. Pan, G.-J. Song, Nonnegative low rank tensor approximations with multidimensional image applications, Numer. Math. 153 (1) (2023) 141–170.
[5] J.-H. Yang, X.-L. Zhao, T.-H. Ma, Y. Chen, T.-Z. Huang, M. Ding, Remote sensing images destriping using unidirectional hybrid total variation and nonconvex low-rank regularization, J. Comput. Appl. Math. 363 (2020) 124–144.
[6] L.-J. Deng, M. Feng, X.-C. Tai, The fusion of panchromatic and multispectral remote sensing images via tensor-based sparse modeling and hyper-Laplacian prior, Inf. Fusion 52 (2019) 76–89.
[7] L. Zhuang, M.K. Ng, L. Gao, Z. Wang, Eigen-CNN: Eigenimages plus eigennoise level maps guided network for hyperspectral image denoising, IEEE Trans. Geosci. Remote Sens. (2024) http://dx.doi.org/10.1109/TGRS.2024.3379199, 1–1.
[8] J.-T. Sun, H.-J. Zeng, H. Liu, Y. Lu, Z. Chen, Cubesvd: a novel approach to personalized web search, in: Proceedings of the 14th International Conference on World Wide Web, 2005, pp. 382–390.
[9] D.A. Lima, G.M. Oliveira, A cellular automata ant memory model of foraging in a swarm of robots, Appl. Math. Model. 47 (2017) 551–572.
[10] T.G. Kolda, B.W. Bader, J.P. Kenny, Higher-order web link analysis using multilinear algebra, in: Fifth IEEE International Conference on Data Mining, ICDM'05, IEEE, 2005, pp. 8–pp.
[11] V.N. Varghees, M.S. Manikandan, R. Gini, Adaptive MRI image denoising using total-variation and local noise estimation, in: IEEE-International Conference on Advances in Engineering, Science and Management, ICAESM-2012, IEEE, 2012, pp. 506–511.
[12] N. Kreimer, M.D. Sacchi, A tensor higher-order singular value decomposition for prestack seismic data noise reduction and interpolation, Geophysics 77 (3) (2012) V113–V122.
[13] E. Candes, B. Recht, Exact matrix completion via convex optimization, Commun. ACM 55 (6) (2012) 111–119.
[14] E.J. Candès, X. Li, Y. Ma, J. Wright, Robust principal component analysis? J. ACM 58 (3) (2011) 1–37.
[15] Z. Song, D. Woodruff, H. Zhang, Sublinear time orthogonal tensor decomposition, Adv. Neural Inf. Process. Syst. 29 (2016).
[16] N. Li, B. Li, Tensor completion for on-board compression of hyperspectral images, in: IEEE International Conference on Image Processing, IEEE, 2010, pp. 517–520.
[17] M.G. Asante-Mensah, A.H. Phan, S. Ahmadi-Asl, Z. Al Aghbari, A. Cichocki, Image reconstruction using superpixel clustering and tensor completion, Signal Process. 212 (2023) 109158.
[18] J.D. Carroll, J.-J. Chang, Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition, Psychometrika 35 (3) (1970) 283–319.
[19] J. Douglas Carroll, S. Pruzansky, J.B. Kruskal, CANDELINC: A general approach to multidimensional analysis of many-way arrays with linear constraints on parameters, Psychometrika 45 (1980) 3–24.
[20] L.R. Tucker, Some mathematical notes on three-mode factor analysis, Psychometrika 31 (3) (1966) 279–311.
[21] S. Gandy, B. Recht, I. Yamada, Tensor completion and low-n-rank tensor recovery via convex optimization, Inverse Probl. 27 (2) (2011) 025010.
[22] C.J. Hillar, L.-H. Lim, Most tensor problems are NP-hard, J. ACM 60 (6) (2013) 1–39.
[23] K. Braman, Third-order tensors as linear operators on a space of matrices, Linear Algebra Appl. 433 (7) (2010) 1241–1253.
[24] M.E. Kilmer, C.D. Martin, Factorization strategies for third-order tensors, Linear Algebra Appl. 435 (3) (2011) 641–658.

[25] M.E. Kilmer, K. Braman, N. Hao, R.C. Hoover, Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging, SIAM J. Matrix Anal. Appl. 34 (1) (2013) 148–172.
[26] X. Cheng, W. Kong, X. Luo, W. Qin, F. Zhang, J. Wang, Tensor completion via joint reweighted tensor Q-nuclear norm for visual data recovery, Signal Process. 219 (2024) 109407.
[27] V. Saragadam, R. Balestriero, A. Veeraraghavan, R.G. Baraniuk, Deeptensor: Low-rank tensor decomposition with deep network priors, IEEE Trans. Pattern Anal. Mach. Intell. (2024).
[28] C.D. Martin, R. Shafer, B. LaRue, An order-p tensor factorization with applications in imaging, SIAM J. Sci. Comput. 35 (1) (2013) A474–A490.
[29] E. Kernfeld, M. Kilmer, S. Aeron, Tensor–tensor products with invertible linear transforms, Linear Algebra Appl. 485 (2015) 545–570.
[30] W.-H. Xu, X.-L. Zhao, M. Ng, A fast algorithm for cosine transform based tensor singular value decomposition, 2019, arXiv preprint arXiv:1902.03070.
[31] C. Lu, X. Peng, Y. Wei, Low-rank tensor completion with a new tensor nuclear norm induced by invertible linear transforms, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 5996–6004.
[32] G. Song, M.K. Ng, X. Zhang, Robust tensor completion using transformed tensor singular value decomposition, Numer. Linear Algebra Appl. 27 (3) (2020) e2299.
[33] T.-X. Jiang, M.K. Ng, X.-L. Zhao, T.-Z. Huang, Framelet representation of tensor nuclear norm for third-order tensor completion, IEEE Trans. Image Process. 29 (2020) 7233–7244.
[34] T.-X. Jiang, X.-L. Zhao, H. Zhang, M.K. Ng, Dictionary learning with low-rank coding coefficients for tensor completion, IEEE Trans. Neural Netw. Learn. Syst. 34 (2) (2023) 932–946.
[35] Y.-S. Luo, X.-L. Zhao, T.-X. Jiang, Y. Chang, M.K. Ng, C. Li, Self-supervised nonlinear transform-based tensor nuclear norm for multi-dimensional image recovery, IEEE Trans. Image Process. 31 (2022) 3793–3808.
[36] W. Zhang, Y. Yang, Z. Li, Q.M.J. Wu, Progressive learning model for big data analysis using subnetwork and Moore-Penrose inverse, IEEE Trans. Multimed. 26 (2024) 8104–8118, http://dx.doi.org/10.1109/TMM.2024.3375774.
[37] W. Zhang, Q.J. Wu, Y. Yang, T. Akilan, Multimodel feature reinforcement framework using Moore–Penrose inverse for big data analysis, IEEE Trans. Neural Netw. Learn. Syst. 32 (11) (2020) 5008–5021.
[38] W. Zhang, Q.M.J. Wu, W.G.W. Zhao, H. Deng, Y. Yang, Hierarchical one-class model with subnetwork for representation learning and outlier detection, IEEE Trans. Cybern. 53 (10) (2023) 6303–6316, http://dx.doi.org/10.1109/TCYB.2022.3166349.
[39] Z. Zhang, S. Aeron, Exact tensor completion using t-SVD, IEEE Trans. Signal Process. 65 (6) (2016) 1511–1526.
[40] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, S. Yan, Tensor robust principal component analysis: Exact recovery of corrupted low-rank tensors via convex optimization, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 5249–5257.
[41] Z. Zhang, S. Aeron, Exact tensor completion using t-SVD, IEEE Trans. Signal Process. 65 (6) (2017) 1511–1526.
[42] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification, in: Proceedings of the IEEE International Conference on Computer Vision, ICCV, 2015.
[43] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.
[44] Z. Wang, A.C. Bovik, H.R. Sheikh, E. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Trans. Signal Process. 13 (4) (2004) 600–612.
[45] B.R. Shivakumar, S.V. Rajashekararadhya, Performance évaluation of spectral angle mapper and spectral correlation mapper classifiers over multiple remote sensor data, in: 2017 Second International Conference on Electrical, Computer and Communication Technologies, ICECCT, 2017, pp. 1–6, http://dx.doi.org/10.1109/ICECCT.2017.8117946.
[46] J. Liu, P. Musialski, P. Wonka, J. Ye, Tensor completion for estimating missing values in visual data, IEEE Trans. Pattern Anal. Mach. Intell. 35 (1) (2013) 208–220, http://dx.doi.org/10.1109/TPAMI.2012.39.
[47] Z. Zhang, G. Ely, S. Aeron, N. Hao, M. Kilmer, Novel methods for multilinear data completion and de-noising based on tensor-SVD, in: IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 3842–3849, http://dx.doi.org/10.1109/CVPR.2014.485.
[48] Z.-C. Wu, T.-Z. Huang, L.-J. Deng, H.-X. Dou, D. Meng, Tensor wheel decomposition and its tensor completion application, Adv. Neural Inf. Process. Syst. 35 (2022) 27008–27020.
[49] H. Wang, J. Peng, W. Qin, J. Wang, D. Meng, Guaranteed tensor recovery fused low-rankness and smoothness, IEEE Trans. Pattern Anal. Mach. Intell. (2023).
[50] Y. Luo, X.-L. Zhao, D. Meng, T.-X. Jiang, HLRTF: Hierarchical low-rank tensor factorization for inverse problems in multi-dimensional imaging, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2022, pp. 19303–19312.

Xi-Zhuo Li received the B.S. degree in Computer Science and Technology from Southwestern University of Finance and Economics of China, Chengdu, China, in 2022. He is currently pursuing the M.S. degree with the School of Computing and Artificial Intelligence, Southwestern University of Finance and Economics of China, Chengdu, China. His research interests include model-based tensor modeling and unsupervised learning for low-level visual tasks.

Tai-Xiang Jiang received the Ph.D. degrees in mathematics from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, 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, tensor decomposition and multi-dimensional image processing.

Liqiao Yang received the M.S. degree in mathematics from University of Electronic Science and Technology of China, Chengdu, in 2020, and received the Ph.D. degrees in mathematics from the University of Macau, China, in 2024. She is currently a postdoctoral researcher at the Southwestern University of Finance and Economics. Her current research interests include quaternion modeling and algorithms for high-order data recovery.

Guisong Liu received the B.S. degree in mechanics from Xi'an Jiao Tong University, Xi'an, China, in 1995, and the M.S. degree in automatics and the Ph.D. degree in computer science from the University of Electronic Science and Technology of China, Chengdu, China, in 2000 and 2007, respectively. He was a Visiting Scholar with Humbolt University, Berlin, Germany, in 2015. Before 2021, he was a Professor with the School of Computer Science and Engineering, University of Electronic Science and Technology of China. He is currently a Professor and the Dean of the School of Computing and Artificial Intelligence, Southwestern University of Finance and Economics, Chengdu. He has filed over 20 patents, and published over 70 scientific conference and journal papers. His research interests include pattern recognition, neural networks, and machine learning.