HLRTF: Hierarchical Low-Rank Tensor Factorization for Inverse Problems in Multi-Dimensional Imaging

Yisi Luo[1]   Xile Zhao[1*]   Deyu Meng[2,3]   Taixiang Jiang[4]

[1]University of Electronic Science and Technology of China, Chengdu, China
[2]Xi'an Jiaotong University, Xi'an, China
[3]Peng Cheng Laboratory, Shenzhen, China
[4]Southwestern University of Finance and Economics, Chengdu, China

yisiluo1221@foxmail.com, xlzhao122003@163.com, dymeng@mail.xjtu.edu.cn, taixiangjiang@gmail.com

Abstract

Inverse problems in multi-dimensional imaging, e.g., completion, denoising, and compressive sensing, are challenging owing to the big volume of the data and the inherent ill-posedness. To tackle these issues, this work unsupervisedly learns a hierarchical low-rank tensor factorization (HLRTF) by solely using an observed multi-dimensional image. Specifically, we embed a deep neural network (DNN) into the tensor singular value decomposition framework and develop the HLRTF, which captures the underlying low-rank structures of multi-dimensional images with compact representation abilities. This DNN herein serves as a nonlinear transform from a vector to another to help obtain a better low-rank representation. Our HLRTF infers the parameters of the DNN and the underlying low-rank structure of the original data from its observation via the gradient descent using a non-reference loss function in an unsupervised manner. To address the vanishing gradient in extreme scenarios, e.g., structural missing pixels, we introduce a parametric total variation regularization to constrain the DNN parameters and the tensor factor parameters with theoretical analysis. We apply our HLRTF for typical inverse problems in multi-dimensional imaging including completion, denoising, and snapshot spectral imaging, which demonstrates its generality and wide applicability. Extensive results illustrate the superiority of our method as compared with state-of-the-art methods.

1. Introduction

Tensor factorization family methods extend familiar matrix cases to multi-dimensional modalities for effective analysis and processing of real-world multi-dimensional images, e.g., videos, hyperspectral images (HSIs), and
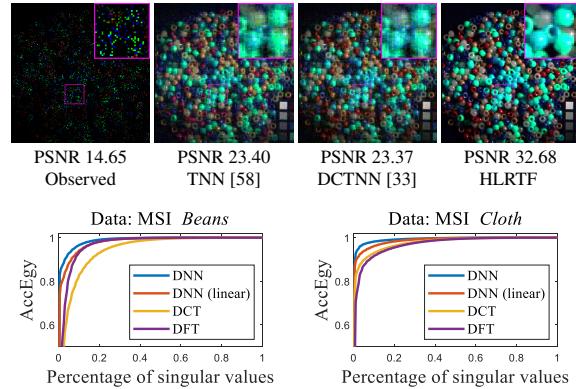

Figure 1. Top: The results of multi-dimensional image completion by TNN (based on the DFT) [58], DCTNN (based on the DCT) [33], and HLRTF (based on the DNN transform) on MSI *Beans* with sampling rate 0.1. Bottom: The AccEgy (AccEgy $= \sum_{i=1}^{k} \sigma_i^2 / \sum_j \sigma_j^2$ with σ_i denotes the i-th singular value [46]) versus the percentage of singular values of the transformed frontal slices by different transforms (i.e., the DNN transform, the DNN transform with only one layer, the DCT, and the DFT). The DNN transform obtains lower-rank transformed frontal slices and thus the corresponding HLRTF achieves better recovery performance.

multispectral images (MSIs) [1, 11]. Real-world multi-dimensional images are usually self-correlated and thus enjoy intrinsic low-rank structures [15, 28]. Therefore, low-rank tensor factorization methods utilize this property to design specific forms and operations to exploit the low-rankness or/and enhance the low-rankness by minimizing the nuclear norm [17, 52, 60, 61], successfully applying to various applications such as hyperspectral imaging [41,48], image/video inpainting [26], network compression [42], and recommender system [6, 7].

Most tensor factorization methods rely on multilinear operations. The classic Tucker decomposition and CP decomposition [13, 24, 28, 44] were proposed for tensor analysis. Recently, tensor network decompositions including

---
[*]Corresponding author

tensor train decomposition [39], tensor ring decomposition [14], and fully connected tensor network decomposition [60] were studied. In this paper, we focus on the tensor singular value decomposition (t-SVD) [22], which is based on the tensor-tensor product (t-product) [20,21]. The t-SVD extended matrix SVD to tensor cases without flattening and information loss inside tensor modalities. Based on the t-SVD, the tensor tubal rank [21,32] was defined. Its convex relaxation, the tensor nuclear norm (TNN), was studied and applied to low-rank tensor recovery [18,25,32,33,58]. More recently, the low-tubal-rank tensor factorization [30,61] was proposed by factorizing a tensor into two smaller tensors through the t-product, preserving the low-tubal-rankness of the tensor and avoiding the computing of the SVD for faster implementations.

Nonetheless, the tensor tubal-rank is based on a linear transform (e.g., the discrete Fourier transform (DFT) [32] or the discrete cosine transform (DCT) [33]). The linear transform is applied on the tubes of a tensor to transform it into a low-rank representation, then the matrix rank of the transform frontal slices is considered to define the tubal-rank, see Definition 2. Considering the complex and diversified topology structures of real-world data, it is highly possible that the transform between the original tensor and the optimal low-rank representation is nonlinear and hierarchical, which can not be interpreted by the linear transform.

In this paper, we propose to replace the linear transform with a deep neural network (DNN), which consists of multiple linear layers and nonlinear activation functions. The motivation is that the DNN transform can access much lower-rank transformed frontal slices and hence a better low-rank representation can be obtained. To validate this, we plot the AccEgy with respect to the percentage of singular values of the transformed frontal slices in Fig. 1. We can observe that the energy obtained by the DNN transform is concentrated in larger singular values, which reveals that a compact low-rank representation is obtained. Thus, the recovery performance can be reasonably improved.

Equipped with the DNN transform, we deduce a new tensor rank called the hierarchical-tubal-rank. Correspondingly, we propose the hierarchical low-rank tensor factorization (HLRTF) and prove its capability to capture the low-rank structure. The HLRTF factorizes a tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ into $\mathcal{X} = \mathcal{A} *_f \mathcal{B}$, where $\mathcal{A} \in \mathbb{R}^{n_1 \times r \times n_3}$ and $\mathcal{B} \in \mathbb{R}^{r \times n_2 \times n_3}$ are two smaller tensors and $*_f$ is the t-product induced by the DNN $f(\cdot)$ (see Definition 4). We further develop an equivalent form of HLRTF, i.e., $\mathcal{X} = g(\hat{\mathcal{A}} \triangle \hat{\mathcal{B}})$, where $g(\cdot)$ is the inverse DNN of $f(\cdot)$, $\triangle$ is the face-wise product [20], and $\hat{\mathcal{A}}, \hat{\mathcal{B}}$ are two smaller tensor factors, to reduce computational expense. To tackle inverse problems in multi-dimensional imaging, we simultaneously optimize the tensor factors $\hat{\mathcal{A}}, \hat{\mathcal{B}}$ and learn the inverse DNN $g(\cdot)$ using the fidelity loss between $g(\hat{\mathcal{A}} \triangle \hat{\mathcal{B}})$ and the obser-

vation. In this way, the low-rank $\mathcal{X}$ can be readily obtained though $\mathcal{X} = g(\hat{\mathcal{A}} \triangle \hat{\mathcal{B}})$ in an unsupervised manner.

To face the difficulty that in some extreme situations, our method would suffer from unavoidable vanishing gradient, we propose the parametric total variation (PTV) regularization for DNN parameters and tensor factor parameters, successfully addressing the vanishing gradient. We prove that PTV is an upper bound of the traditional 3D total variation (3DTV) regularization [40], while its computational complexity is much lower than that of 3DTV regularization. Therefore, PTV efficiently captures the local smoothness [40] of multi-dimensional images to enhance the robustness of HLRTF.

We summarize the contributions of this paper as follows:

- By embedding a DNN as a nonlinear transform into the t-SVD framework, we propose the HLRTF to capture the underlying low-rank structure of multi-dimensional images with compact representation abilities. We provide algebraic property of the HLRTF, which leads to solid foundations of its potential capacity. With the loss function tailored corresponding to different observations, high-quality recovery results can be obtained after the DNN parameters and tensor factor parameters are unsupervisedly learned.

- To address extreme structural missing cases, we propose the PTV regularization to constrain the DNN parameters and tensor factor parameters, successfully addressing the vanishing gradient issue. Our analysis shows that PTV is rooted in classic 3DTV and has a lower computational complexity.

- We apply HLRTF to typical inverse problems in multi-dimensional imaging including multi-dimensional image completion, denoising, and snapshot spectral imaging. Extensive experiments validate the generalization abilities of HLRTF for different tasks and its superior performance over state-of-the-art methods.

1.1. Related Work

1.1.1 Transform Induced t-SVD

In the literature, there were other transforms used in the t-SVD. The initial one was the DFT [32, 58]. The DCT was employed for real arithmetic computation [35]. Some unitary transforms were adopted to obtain a lower-tubal-rank tensor [37, 45]. Lu et al. [33] showed that any invertible linear transforms were able to induce the TNN. Recently, non-invertible transforms were used in t-SVD [18, 19, 25], and some of them were data-dependent [19, 25]. However, these transforms are all linear and their corresponding TNN frameworks require SVD computing. Our DNN transform is a nonlinear transform and the proposed HLRTF is SVD-free, which largely saves computational costs.

1.1.2 Matrix/Tensor Factorization with Deep Learning

Recently, there are growing appeals for combining deep learning techniques and matrix/tensor factorization methods. The deep matrix factorization [2,10,51] attempted integrating DNNs with matrix factorization. The tensor Tucker and CP factorizations were combined into DNNs for effective deep learning [6,7,9,27,38,57]. However, these deep tensor factorization-based methods are mostly supervised learning methods, which may lack generalization abilities for different tasks.

2. Notations and Preliminaries

Matrices and tensors are denoted as $\mathbf{X}$, $\mathcal{X}$, respectively. Given a tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, $\mathcal{X}(i,j,k)$ denotes the i, j, k-th element of $\mathcal{X}$ and $\mathcal{X}^{(i)} \in \mathbb{R}^{n_1 \times n_2}$ denotes the i-th frontal slice of $\mathcal{X}$. $\times_3$ denotes the mode-3 tensor-matrix product [24], i.e., $\mathcal{X} \times_3 \mathbf{A} = \texttt{fold}(\mathbf{A}\texttt{unfold}(\mathcal{X}))$, where $\texttt{unfold}(\cdot) : \mathbb{R}^{n_1 \times n_2 \times n_3} \to \mathbb{R}^{n_3 \times n_1 n_2}$ is the unfolding operator and $\texttt{fold}(\cdot)$ is its inverse operator. $\triangle$ denotes the face-wise product between two tensors [20], i.e., $\mathcal{C} = \mathcal{A} \triangle \mathcal{B} \Leftrightarrow \mathcal{C}^{(i)} = \mathcal{A}^{(i)} \mathcal{B}^{(i)}$. ∇_x, ∇_y, and ∇_z respectively denote the vertical, horizontal, and temporal/spectral derivative operators of a tensor [1] [34].

Definition 1. (T-product) *[21] The tensor-tensor product between $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{B} \in \mathbb{R}^{n_2 \times n_4 \times n_3}$ is defined as $\mathcal{A} * \mathcal{B} = ((\mathcal{A} \times_3 \mathbf{F}) \triangle (\mathcal{B} \times_3 \mathbf{F})) \times_3 \mathbf{F}^{-1} \in \mathbb{R}^{n_1 \times n_4 \times n_3}$, where $\mathbf{F} \in \mathbb{R}^{n_3 \times n_3}$ is the DFT matrix and $\mathbf{F}^{-1}$ is the inverse DFT matrix.*

Definition 2. (Tensor tubal-rank) *[21] The tensor tubal rank of $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is defined as*

$$\text{rank}_t(\mathcal{A}) \triangleq \max_{i=1,2,\cdots,n_3} \{\text{rank}((\mathcal{A} \times_3 \mathbf{F})^{(i)})\}, \quad (1)$$

where $\mathbf{F} \in \mathbb{R}^{n_3 \times n_3}$ is the DFT matrix.

Theorem 1. (Low-tubal-rank tensor factorization) *[30, 61] Let $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, $\mathcal{Y} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, and $\mathcal{Z} \in \mathbb{R}^{n_2 \times n_4 \times n_3}$ be arbitrary tensors, then*

(i) *If $\text{rank}_t(\mathcal{X}) = r$, then there exist two tensors $\mathcal{A} \in \mathbb{R}^{n_1 \times r \times n_3}$ and $\mathcal{B} \in \mathbb{R}^{r \times n_2 \times n_3}$ such that $\mathcal{X} = \mathcal{A} * \mathcal{B}$ hold and they meet $\text{rank}_t(\mathcal{A}) = \text{rank}_t(\mathcal{B}) = r$.*

(ii) $\text{rank}_t(\mathcal{Y} * \mathcal{Z}) \leq \min\{\text{rank}_t(\mathcal{Y}), \text{rank}_t(\mathcal{Z})\}$.

3. Main Results

3.1. Hierarchical Low-Rank Tensor Factorization

In this section, we introduce the proposed HLRTF. Our method is based on the tensor tubal-rank. The tensor tubal-rank considers the interactions of frontal slices by connecting all frontal slices of $\mathcal{A}$ with the linear DFT along the

third mode. Such interactions are ubiquitous in real world data, e.g., the temporal correlations of videos and the spectral correlations of MSIs. Given a degraded tensor, the basic assumption is that the underlying tensor is low-tubal-rank, and therefore tubal-rank minimization can be used to recover the underlying tensor [30, 61].

However, considering the complex and diverse structures of real-world multi-dimensional images, it is highly possible that the linear DFT can not transform the original tensor into a desirable low-rank representation [8]. Thus, we propose to replace the DFT with a DNN, which consists of multiple linear layers and nonlinear functions. The DNN is expected to capture the nonlinear interactions inside data to obtain a better low-rank representation, see Fig. 1. Equipped with the DNN, we deduce a new tensor rank called the hierarchical tubal-rank.

Definition 3. (Hierarchical tubal-rank) *Given a tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, matrices $\{\mathbf{W}_j \in \mathbb{R}^{n_3 \times n_3}\}_{j=1}^{k}$, and a nonlinear scalar function $\sigma(\cdot)$, the hierarchical tubal-rank is defined as*

$$\text{rank}_h(\mathcal{A}) \triangleq \max_{i=1,2,\cdots,n_3} \{\text{rank}(f(\mathcal{A})^{(i)})\}, \quad (2)$$

where $f(\mathcal{A}) \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is given by $f(\mathcal{A}) = \sigma(\cdots\sigma(\sigma(\mathcal{A} \times_3 \mathbf{W}_1) \times_3 \mathbf{W}_2) \cdots \times_3 \mathbf{W}_{k-1}) \times_3 \mathbf{W}_k$.

In Definition 3, $f(\cdot)$ is a DNN with k layers. Except the last layer, all layers are followed by the nonlinear activation function $\sigma(\cdot)$ (In this work, $\sigma(\cdot)$ is set as the invertible LeakyReLU function [12]). The DNN more effectively captures the nonlinear interactions inside data for a better low-rank representation than that of linear transforms (see the AccEgy in Fig. 1). In fact, if we let $k = 1$, then $f(\cdot)$ degrades to a linear transform, i.e., the linear transform [19, 25, 33] is just a special case of $f(\cdot)$.

The direct way to minimize the hierarchical tubal-rank is to minimize the nuclear norm of the transformed frontal slices [32]. However, minimizing the nuclear norm, which needs to compute the SVD, is time-consuming [61]. Thus, we turn to formulate a hierarchical low-rank tensor factorization method that implicitly preserves the hierarchical tubal-rank without SVD computing. As preparation, we study the invertibility of the DNN $f(\cdot)$ and define a new t-product induced by $f(\cdot)$.

Lemma 1. (Invertibility) *Suppose that $\{\mathbf{W}_j \in \mathbb{R}^{n_3 \times n_3}\}_{j=1}^{k}$ are full-rank and $\sigma(\cdot)$ is invertible. Then, there exists a DNN with k layers, denoted as $g(\cdot)$, such that $g(f(\mathcal{A})) = f(g(\mathcal{A})) = \mathcal{A}$ holds for arbitrary $\mathcal{A}$.*

Proof. Let $\sigma^{-1}(\cdot)$ be the inverse function of $\sigma(\cdot)$ and $\mathbf{H}_j \in \mathbb{R}^{n_3 \times n_3}$ ($j = 1, 2, \cdots, n_3$) be the inverse matrix of $\mathbf{W}_{n_3-j+1}$. Then, $g(f(\mathcal{A})) = f(g(\mathcal{A})) = \mathcal{A}$ holds for arbitrary $\mathcal{A}$, where $g(\mathcal{A}) = \sigma^{-1}(\cdots\sigma^{-1}(\sigma^{-1}(\mathcal{A} \times_3 \mathbf{H}_1) \times_3 \mathbf{H}_2) \cdots \times_3 \mathbf{H}_{k-1}) \times_3 \mathbf{H}_k$. $\square$

---

[1] The derivative operators can be also applied to matrices since a matrix $\mathbf{X} \in \mathbb{R}^{n_1 \times n_2}$ can be viewed as a tensor of size $n_1 \times n_2 \times 1$.

Definition 4. (T-product induced by $f(\cdot)$) *Given $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{B} \in \mathbb{R}^{n_2 \times n_4 \times n_3}$, the t-product induced by $f(\cdot)$ is defined as $\mathcal{A} *_f \mathcal{B} = g(f(\mathcal{A}) \triangle f(\mathcal{B})) \in \mathbb{R}^{n_1 \times n_4 \times n_3}$, where $g(\cdot)$ is the inverse transform of $f(\cdot)$.*

Theorem 2. (Hierarchical low-rank tensor factorization) *Let $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, $\mathcal{Y} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, $\mathcal{Z} \in \mathbb{R}^{n_2 \times n_4 \times n_3}$ be arbitrary tensors, $f(\cdot)$ be a DNN, and the suppositions in Lemma 1 hold. Then*

(i) *If $\mathrm{rank}_h(\mathcal{X}) = r$, then there exist two tensors $\mathcal{A} \in \mathbb{R}^{n_1 \times r \times n_3}$ and $\mathcal{B} \in \mathbb{R}^{r \times n_2 \times n_3}$ such that $\mathcal{X} = \mathcal{A} *_f \mathcal{B}$ hold and they meet $\mathrm{rank}_h(\mathcal{A}) = \mathrm{rank}_h(\mathcal{B}) = r$.*

(ii) $\mathrm{rank}_h(\mathcal{Y} *_f \mathcal{Z}) \leq \min\{\mathrm{rank}_h(\mathcal{Y}), \mathrm{rank}_h(\mathcal{Z})\}$.

By utilizing Theorem 2, we can characterize the low-rankness of a tensor by factorizing it as $\mathcal{X} = \mathcal{A} *_f \mathcal{B}$, where $\mathcal{A} \in \mathbb{R}^{n_1 \times r \times n_3}$, $\mathcal{B} \in \mathbb{R}^{r \times n_2 \times n_3}$, and $\mathrm{rank}_h(\mathcal{X}) \leq r$ holds. By changing r (which can be efficiently done by changing the size of $\mathcal{A}, \mathcal{B}$), we can control the hierarchical tubal-rank of $\mathcal{X}$ for low-rank tensor recovery. When $k = 1$, the proposed HLRTF can degrade to the low-tubal-rank tensor factorization in Theorem 1 [61], i.e., the low-tubal-rank tensor factorization is a special case of our HLRTF.

Based on Definition 4, instead of learning $f(\cdot)$ and calculate its inverse $g(\cdot)$ to obtain the low-rank tensor $\mathcal{X} = \mathcal{A} *_f \mathcal{B}$, we can cleverly start from the latent tensor factors $\hat{\mathcal{A}} \triangleq f(\mathcal{A})$ and $\hat{\mathcal{B}} \triangleq f(\mathcal{B})$ to reformulate the HLRTF as $\mathcal{X} = g(\hat{\mathcal{A}} \triangle \hat{\mathcal{B}})$. Thus, we only need to learn the inverse $g(\cdot)$ and tensor factors $\hat{\mathcal{A}}$ and $\hat{\mathcal{B}}$ to obtain $\mathcal{X}$, which avoids the complicated calculation of the inverse $g(\cdot)$ from $f(\cdot)$.

3.2. Multi-Dimensional Image Recovery Models

The optimization model of our HLRTF for multi-dimensional image recovery is formulated as

$$\min_{\hat{\mathcal{A}}, \hat{\mathcal{B}}, \{\mathbf{H}_j\}_{j=1}^k} L(g(\hat{\mathcal{A}} \triangle \hat{\mathcal{B}}), \mathcal{O}), \qquad (3)$$

where $\mathcal{O} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is the observation. $\hat{\mathcal{A}} \in \mathbb{R}^{n_1 \times r \times n_3}$ and $\hat{\mathcal{B}} \in \mathbb{R}^{r \times n_2 \times n_3}$ are tensor factors and $g(\cdot)$ is a DNN with parameters $\{\mathbf{H}_j\}_{j=1}^k$. L denotes the fidelity loss between our result and the observation, and flexibly varies for different tasks. Three tasks are considered in this work:

- **Multi-dimensional image completion** [18], which aims at recovering an underlying low-rank tensor from its incompleted observation. The corresponding fidelity term is $L(\mathcal{X}, \mathcal{O}) = \|(\mathcal{X} - \mathcal{O})_\Omega\|_F^2$, where Ω is the support of observed entries.

- **Multispectral image denoising** [48], which aims at recovering a clean multi-dimensional image from its noisy observation. We consider mixed noise including Gaussian noise, impulse noise, and stripe deadlines [29]. The fidelity term is $L(\mathcal{X}, \mathcal{O}) = \|\mathcal{X} - \mathcal{O}\|_{\ell_1}$.

- **Snapshot spectral imaging** [31], which aims at recovering the underlying multi-dimensional image from its low-dimensional measurement and binary masks. The fidelity term is $L(\mathcal{X}, \mathbf{O}) = \|\sum_{i=1}^{n_3} \mathcal{C}^{(i)} \odot \mathcal{X}^{(i)} - \mathbf{O}\|_F^2$, where $\odot$ is the element-wise product. Here, $\mathbf{O} \in \mathbb{R}^{n_1 \times n_2}$ is the measurement and $\{\mathcal{C}^{(i)}\}_{i=1}^{n_3}$ denote n_3 masks, which are given in advance.

We remark here that our HLRTF characterizes the low-rank structure of multi-dimensional images with compact representation abilities. Thus, it is not limited to the above tasks. For other tasks, with suitable formulations of L, our HLRTF is believed to perform well.

3.3. Optimization

We directly use off-the-shelf gradient descent-based algorithms to optimize the non-convex problem (3). Specifically, we optimize $\boldsymbol{\Theta} = \{\hat{\mathcal{A}}, \hat{\mathcal{B}}, \{\mathbf{H}_j\}_{j=1}^k\}$ using the efficient adaptive moment estimation (Adam) algorithm [23], in which the gradient $\frac{\partial L}{\partial \boldsymbol{\Theta}}$ is used in back propagation to minimize $L(g(\hat{\mathcal{A}} \triangle \hat{\mathcal{B}}), \mathcal{O})$. Benefit from our elegant design of importing the hierarchical structure in the transform pipeline within the t-SVD framework and directly learning the inverse DNN $g(\cdot)$, the holistic computational burden is low as we only need to infer k matrices, i.e., $\{\mathbf{H}_j\}_{j=1}^k$, and two smaller tensors, i.e., $\hat{\mathcal{A}}$ and $\hat{\mathcal{B}}$.

However, in our unsupervised framework, a tricky issue occurs in some challenging cases. Next, we first clarify the difficulty we faced and then propose the PTV to fix it.

3.3.1 Vanishing Gradient

We take the completion task as an example. When some slices of the multi-dimensional image are missing, as shown in Fig. 2, directly minimizing (3) leads to poor results. Digging into back propagation steps, we find that the gradient would vanish in this situation.

Lemma 2. (Vanishing gradient) *Take the tensor completion task as an example. The fidelity term is $L(\mathcal{X}, \mathcal{O}) = \|(\mathcal{X} - \mathcal{O})_\Omega\|_F^2$, where $\mathcal{X} = g(\hat{\mathcal{A}} \triangle \hat{\mathcal{B}})$. Then*
(i) *If the i-th horizontal slice of $\mathcal{O}$ is missed, i.e., $(i, b, c) \notin \Omega$ for arbitrary b, c, then the following equalities hold:*

$$\frac{\partial L(\mathcal{X}, \mathcal{O})}{\partial \hat{\mathcal{A}}(i, v, w)} = 0, \; v = 1, 2, \cdots, r, \; w = 1, 2, \cdots, n_3. \quad (4)$$

(ii) *If the i-th lateral slice of $\mathcal{O}$ is missed, i.e., $(a, i, c) \notin \Omega$ for arbitrary a, c, then the following equalities hold:*

$$\frac{\partial L(\mathcal{X}, \mathcal{O})}{\partial \hat{\mathcal{B}}(u, i, w)} = 0, \; u = 1, 2, \cdots, r, \; w = 1, 2, \cdots, n_3. \quad (5)$$

(iii) *If the i-th frontal slice of $\mathcal{O}$ is missed, i.e., $(a, b, i) \notin \Omega$ for arbitrary a, b, then the following equalities hold:*

$$\frac{\partial L(\mathcal{X}, \mathcal{O})}{\partial \mathbf{H}_k(i, v)} = 0, \; v = 1, 2, \cdots, n_3. \quad (6)$$

PSNR 39.65 | PSNR 41.26 | PSNR 40.07 | PSNR 42.79 | PSNR 43.11

PSNR 18.20 | PSNR 21.53 | PSNR 21.84 | PSNR 20.91 | PSNR 26.36

PSNR 16.25 No Reg. | PSNR 15.94 $\|\nabla_x \hat{\mathcal{A}}\|_{\ell_1}$ | PSNR 15.95 $\|\nabla_y \hat{\mathcal{B}}\|_{\ell_1}$ | PSNR 39.79 $\|\nabla_x \mathbf{H}\|_{\ell_1}$ | PSNR 42.10 PTV($\boldsymbol{\Theta}$)
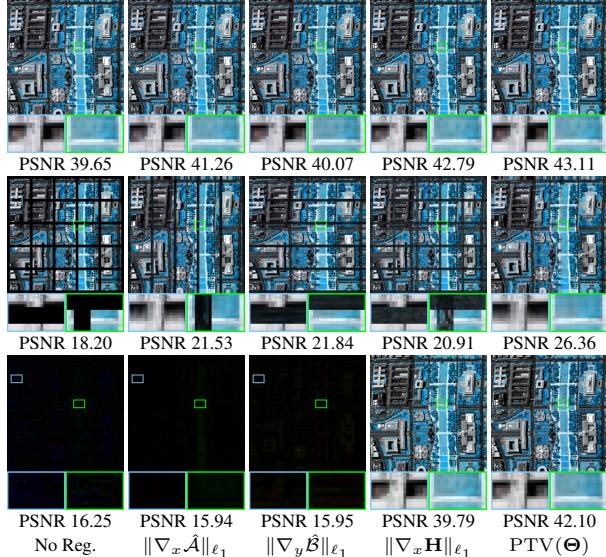
Figure 2. The results of multi-dimensional image completion on HSI *WDC mall* with random missing (first row), horizontal/lateral slice missing (second row), and frontal slice missing (third row) by HLRTF with different regularizations.

Fig. 3 provides a vivid illustration of Lemma 2. Similarly, in other tasks (denoising and snapshot spectral imaging), the vanishing gradient would also occur in extreme conditions. In these conditions, the gradients on some of the parameters in $\boldsymbol{\Theta}$ are zeros and these parameters would never update (see the first column of Fig. 2). This issue is different from the conventional vanishing gradient [3], since we only have one observation and all the parameters in $\boldsymbol{\Theta}$ are inferred in an unsupervised manner. This prevents us to resort to some well-known proven techniques, e.g., batch normalization [16], to address the vanishing gradient. Therefore, new techniques are appealed for.

3.3.2 Parametric Total Variation

To address the vanishing gradient issue, we propose the PTV regularization to constrain the DNN parameters and tensor factor parameters. Suppose that $\mathcal{X} = g(\hat{\mathcal{A}} \triangle \hat{\mathcal{B}})$ and $\boldsymbol{\Theta} = \{\hat{\mathcal{A}}, \hat{\mathcal{B}}, \{\mathbf{H}_j\}_{j=1}^k\}$. Inspired by the fact that nearby parameters are usually more related [4], we propose the PTV regularization, which is formulated as

$$\mathrm{PTV}(\boldsymbol{\Theta}) \triangleq \|\nabla_x \hat{\mathcal{A}}\|_{\ell_1} + \|\nabla_y \hat{\mathcal{B}}\|_{\ell_1} + \|\nabla_x \mathbf{H}_k\|_{\ell_1}. \quad (7)$$

In our PTV regularization, three terms $\|\nabla_x \hat{\mathcal{A}}\|_{\ell_1}$, $\|\nabla_y \hat{\mathcal{B}}\|_{\ell_1}$, and $\|\nabla_x \mathbf{H}_k\|_{\ell_1}$ respectively address the vanishing gradient in (4), (5), and (6). Take (4) as an example, where the i-th horizontal slice of $\hat{\mathcal{A}}$ has zero gradients. The regularization $\|\nabla_x \hat{\mathcal{A}}\|_{\ell_1}$ enforces the nearby horizontal slices of $\hat{\mathcal{A}}$ to have local-similarities, and thus the vanishing gradient in the i-th horizontal slice is addressed. Similarly, $\|\nabla_y \hat{\mathcal{B}}\|_{\ell_1}$ and $\|\nabla_x \mathbf{H}_k\|_{\ell_1}$ address the vanishing gradient in (5) and (6).
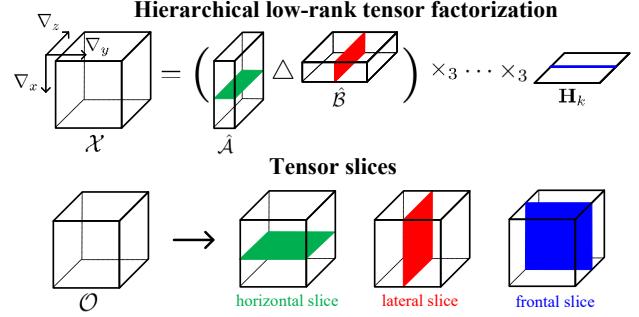

Figure 3. Figurative expressions of Lemma 2. **(i)** If the i-th horizontal slice of $\mathcal{O}$ is missed, then the gradient on the i-th horizontal slice of $\hat{\mathcal{A}}$ equals to zero. **(ii)** If the i-th lateral slice of $\mathcal{O}$ is missed, then the gradient on the i-th lateral slice of $\hat{\mathcal{B}}$ equals to zero. **(iii)** If the i-th frontal slice of $\mathcal{O}$ is missed, then the gradient on the i-th row of $\mathbf{H}_k$ equals to zero.
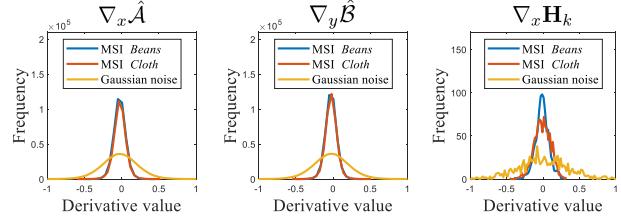

Figure 4. The statistical distributions of derivative values of factors for different data $\mathcal{O}$: the ground-truth MSI *Beans*, the ground-truth MSI *Cloth*, and the Gaussian noise. Given the data $\mathcal{O}$, the factors are obtained by optimizing (3) using the Adam.

We first empirically validate the effectiveness of PTV. In Fig. 4, we display the statistical distributions of the derivative values of $\hat{\mathcal{A}}$, $\hat{\mathcal{B}}$, and $\mathbf{H}_k$. We can observe that when given ground-truth MSIs as the observation $\mathcal{O}$, the derivative values are more concentrated around zero, while they are less so when given random Gaussian noise. This reveals that the factors have local-similarities for ground-truth MSIs, i.e., $\|\nabla_x \hat{\mathcal{A}}\|_{\ell_1}$, $\|\nabla_y \hat{\mathcal{B}}\|_{\ell_1}$, and $\|\nabla_x \mathbf{H}_k\|_{\ell_1}$ tend to have low intensities for real-world multi-dimensional images, which validates the effectiveness of PTV.

Next, we theoretically validate the effectiveness of our PTV by studying the connections between PTV regularization and the traditional 3DTV regularization [40].

Theorem 3. *Suppose that $\mathcal{X} = g(\hat{\mathcal{A}} \triangle \hat{\mathcal{B}})$, where $\{\mathbf{H}_j\}_{j=1}^k$, $\hat{\mathcal{A}}$, and $\hat{\mathcal{B}}$ are bounded and $\sigma^{-1}(\cdot)$ is Lipschitz continuous. Then, there exist three constants $J_1, J_2, J_3 > 0$ such that the following inequalities hold:*

$$\begin{cases} \|\nabla_x \mathcal{X}\|_{\ell_1} \le J_1 \|\nabla_x \hat{\mathcal{A}}\|_{\ell_1} \\ \|\nabla_y \mathcal{X}\|_{\ell_1} \le J_2 \|\nabla_y \hat{\mathcal{B}}\|_{\ell_1} \\ \|\nabla_z \mathcal{X}\|_{\ell_1} \le J_3 \|\nabla_x \mathbf{H}_k\|_{\ell_1}. \end{cases} \quad (8)$$

Let $J = \max\{J_1, J_2, J_3\}$, we have

$$\begin{aligned} \|\mathcal{X}\|_{\mathrm{3DTV}} &\triangleq \|\nabla_x \mathcal{X}\|_{\ell_1} + \|\nabla_y \mathcal{X}\|_{\ell_1} + \|\nabla_z \mathcal{X}\|_{\ell_1} \\ &\le J \, \mathrm{PTV}(\boldsymbol{\Theta}). \end{aligned} \quad (9)$$

Theorem 3 indicates that the traditional 3DTV regularization [40] on $\mathcal{X}$ is upper bounded by the proposed PTV. This reveals that PTV constrains the local smoothness of $\mathcal{X}$, which has been widely considered for real-world multi-dimensional image recovery [47, 48, 59]. Thus, PTV can largely enhance the robustness of HLRTF for various inverse problems in multi-dimensional imaging. Moreover, the computational complexity of PTV is much lower than that of 3DTV. Specifically, it costs $O(r(n_1n_3+n_2n_3)+n_3^2)$ to compute the PTV, while it costs $O(3n_1n_2n_3)$ to compute the 3DTV. Here, $r << \min\{n_1, n_2\}$. Therefore, PTV can efficiently enhance the robustness of HLRTF for multi-dimensional image recovery.

By introducing the PTV in the optimization model, (3) is reformulated as

$$\min_{\hat{\mathcal{A}},\hat{\mathcal{B}},\{\mathbf{H}_j\}_{j=1}^k} L(g(\hat{\mathcal{A}}\triangle\hat{\mathcal{B}}),\mathcal{O}) + \gamma\mathrm{PTV}(\boldsymbol{\Theta}), \qquad (10)$$

where γ is the trade-off parameter. Similarly, we directly use Adam to optimize (10).

4. Experiments

4.1. Ablation Study

We first evaluate the influence of the PTV regularization. We consider multi-dimensional image completion with random missing, horizontal/lateral slice missing, and frontal slice missing cases. We test five approaches: HLRTF without regularizations, HLRTF with $\|\nabla_x\hat{\mathcal{A}}\|_{\ell_1}$ regularization, HLRTF with $\|\nabla_y\hat{\mathcal{B}}\|_{\ell_1}$ regularization, HLRTF with $\|\nabla_x\mathbf{H}\|_{\ell_1}$ regularization, and HLRTF with $\mathrm{PTV}(\boldsymbol{\Theta})$ regularization. The results are shown in Fig. 2. We can observe that all approaches can well recover the HSI with random missing. However, HLRTF without regularizations cannot recover the HSI with slice missing. $\|\nabla_x\hat{\mathcal{A}}\|_{\ell_1}$ can address horizontal slice missing, $\|\nabla_y\hat{\mathcal{B}}\|_{\ell_1}$ can address lateral slice missing, and $\|\nabla_x\mathbf{H}\|_{\ell_1}$ can address frontal slice missing. The $\mathrm{PTV}(\boldsymbol{\Theta})$ combines these three terms and thus can recover HSIs with different slice missing cases. The experimental results well support the analysis in Sec. 3.3.2.

Next, we clarify the influence of the number of layers (i.e., the parameter k) of the DNN. We set $k=1,2,3,4,5$ to test the influence (When $k=1$, the DNN degrades to a linear transform). The results are shown in Table 1. We can observe setting a proper number can improve the performance. However, when k becomes too large, the performance drops, since it is harder to optimize the DNN with deeper structures. Nevertheless, setting k as a moderate number (e.g., $k=2$) can well guarantee the performance.

4.2. Comparisons with State-of-the-Arts

We then compare our method with state-of-the-art methods in three different inverse problems in multi-dimensional

Table 1. The quantitative results for multi-dimensional image completion by HLRTF with different number of layers.

Sampling rate		0.1		0.2		0.3		Time
Data	k	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	(s)
HSIs	$k=1$	30.98	0.930	32.87	0.956	34.33	0.968	355
WDC mall	$k=2$	41.70	0.990	51.41	0.999	54.68	**0.999**	491
($256\times256\times191$)	$k=3$	**45.71**	**0.997**	**52.80**	**0.999**	**56.35**	**0.999**	584
Pavia	$k=4$	45.68	**0.997**	51.77	**0.999**	54.56	**0.999**	598
($200\times200\times80$)	$k=5$	45.14	**0.997**	50.26	**0.999**	52.72	**0.999**	656

imaging: completion [18], denoising [48], and snapshot spectral imaging [31]. We use peak-signal-to-noise-ratio (PSNR) and structural similarity (SSIM) to evaluate the results. Higher PSNR and SSIM values refer to better performance. HLRTF and HLRTF* respectively denote the proposed method with and without the PTV. For implementation details, please refer to supplementary materials.

4.2.1 Datasets and Compared Methods

Multi-Dimensional Image Completion The multi-dimensional image completion aims at recovering the underlying image from its incompleted observation. We adopt two HSIs (*WDC mall* and *Pavia*[2]), two MSIs (*Beans* and *Cloth* [50][3]), and two videos (*Backdoor* and *Yard*[4]) to conduct the experiments. The sampling rates (SRs) of random missing cases are set to 0.1, 0.2, and 0.3. Please refer to supplementary materials for structural slice missing cases. The compared methods include the TNN based on DFT (TNN) [58], the deep matrix factorization (DMF) [10], the low-tubal-rank tensor factorization (TCTF) [61], the TNN based on DCT (DCTNN) [33], the tensor ring decompoistion with rank minimization on latent space (TRLRF) [52], the freamlet transform-based TNN (FTNN) [18], and the fully connectd tensor network decomposition (FCTN) [60].

Multispectral Image Denoising The MSI denoising aims at recovering the clean HSI from its noisy observation. We adopt two HSIs (*WDC mall* and *Pavia*) and four MSIs (*Balloons*, *Fruits* [50], *Pool*, and *Doll*[5]) to conduct the experiments. We consider three noisy cases. **Case 1** includes the Gaussian noise with standard deviation 0.2. **Case 2** includes Gaussian noise with standard deviation 0.2 and impulse noise with SR 0.1. **Case 3** includes the same noise in **Case 2** plus stripe deadline noise in all spectral bands [29]. The compared methods are the low-rank matrix recovery (LRMR) [56], the total variation regularized low-rank tensor decomposition (LRTDTV) [48], the subspace-based

Table 2. The average quantitative results for completion. The **best** and <u>second-best</u> values are highlighted.

Sampling rate		0.1		0.2		0.3		Time
Data	Method	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	(s)
HSIs *WDC mall* (256×256×191) *Pavia* (200×200×80)	TNN	32.69	0.954	38.08	0.984	41.77	0.991	1212
	DMF	20.41	0.476	22.21	0.611	23.78	0.711	1249
	TCTF	19.28	0.413	20.24	0.528	21.34	0.618	504
	DCTNN	36.27	0.979	43.67	0.994	48.68	0.997	761
	TRLRF	28.58	0.895	31.06	0.936	31.84	0.946	2530
	FTNN	37.82	<u>0.984</u>	44.64	0.995	49.05	<u>0.998</u>	4016
	FCTN	40.35	0.982	47.05	<u>0.998</u>	48.99	**0.999**	4829
	HLRTF*	<u>41.11</u>	<u>0.984</u>	<u>47.73</u>	<u>0.998</u>	<u>50.32</u>	**0.999**	407
	HLRTF	**41.70**	**0.990**	**51.41**	**0.999**	**54.68**	**0.999**	491
MSIs *Beans* (256×256×31) *Cloth* (256×256×31)	TNN	24.19	0.796	29.11	0.924	32.82	0.964	254
	DMF	17.35	0.334	19.32	0.495	20.96	0.617	157
	TCTF	17.15	0.349	18.50	0.450	19.95	0.549	109
	DCTNN	24.69	0.814	30.21	0.940	34.49	0.975	190
	TRLRF	25.21	0.818	27.69	0.891	29.54	0.926	1054
	FTNN	26.89	0.887	32.33	0.962	36.27	0.982	977
	FCTN	27.44	0.883	30.85	0.940	32.17	0.955	571
	HLRTF*	<u>28.57</u>	<u>0.916</u>	<u>34.00</u>	<u>0.972</u>	<u>37.24</u>	<u>0.985</u>	328
	HLRTF	**32.13**	**0.964**	**36.71**	**0.986**	**38.77**	**0.991**	370
Videos *Backdoor* (240×320×30) *Yard* (240×320×30)	TNN	30.60	0.928	36.75	0.980	38.07	0.985	294
	DMF	18.64	0.559	32.42	0.963	38.86	<u>0.988</u>	329
	TCTF	18.33	0.563	19.35	0.682	20.79	0.725	124
	DCTNN	33.57	0.966	36.64	0.981	38.07	0.985	202
	TRLRF	32.30	0.952	34.29	0.969	35.11	0.975	754
	FTNN	35.36	0.976	37.78	<u>0.985</u>	39.05	**0.989**	1308
	FCTN	34.94	0.972	37.81	<u>0.985</u>	38.97	<u>0.988</u>	791
	HLRTF*	<u>36.30</u>	<u>0.980</u>	<u>38.32</u>	**0.987**	<u>39.28</u>	**0.989**	331
	HLRTF	**36.69**	**0.982**	**38.45**	**0.987**	**39.30**	**0.989**	377

Table 3. The average quantitative results for MSI denoising. The **best** and <u>second-best</u> values are highlighted.

Case		**Case 1**		**Case 2**		**Case 3**		Time
Data	Method	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	(s)
HSIs *WDC mall* (256×256×191) *Pavia* (200×200×80)	LRMR	28.62	0.901	30.94	0.940	23.64	0.817	226
	LRTDTV	30.16	0.915	32.81	0.953	23.97	0.826	193
	SNLRSF	**34.47**	**0.970**	29.55	0.941	23.41	0.808	755
	D-CNN	29.67	0.919	25.93	0.858	22.26	0.749	829
	SDeCNN	31.94	0.947	26.77	0.892	22.45	0.766	87
	E3DTV	29.76	0.911	33.10	0.959	23.98	0.827	75
	ℓ_0-ℓ_1 HTV	27.04	0.844	29.80	0.911	23.58	0.788	494
	HLRTF*	31.40	0.941	<u>33.50</u>	<u>0.961</u>	24.16	<u>0.836</u>	67
	HLRTF	<u>32.75</u>	<u>0.954</u>	**34.00**	**0.969**	**30.69**	**0.936**	73
MSIs *Balloons* (256×256×31) *Fruits* (256×256×31)	LRMR	28.36	0.779	31.32	0.866	25.14	0.822	74
	LRTDTV	34.22	0.941	34.22	0.914	<u>25.55</u>	0.852	61
	SNLRSF	33.61	0.917	27.16	0.719	23.62	0.653	901
	D-CNN	29.59	0.864	24.25	0.599	22.70	0.564	382
	SDeCNN	<u>34.80</u>	<u>0.942</u>	25.29	0.656	22.85	0.594	24
	E3DTV	31.86	0.928	33.71	0.931	25.43	<u>0.860</u>	18
	ℓ_0-ℓ_1 HTV	32.35	0.907	<u>35.13</u>	<u>0.938</u>	25.48	0.838	128
	HLRTF*	31.41	0.887	32.18	0.894	24.49	0.727	23
	HLRTF	**35.19**	**0.945**	**35.89**	**0.953**	**34.06**	**0.943**	27
MSIs *Pool* (256×288×49) *Doll* (256×288×49)	LRMR	30.20	0.824	33.37	0.921	24.76	0.828	127
	LRTDTV	31.47	<u>0.943</u>	33.31	**0.957**	24.61	<u>0.855</u>	94
	SNLRSF	<u>34.43</u>	0.942	29.59	0.877	23.94	0.779	1035
	D-CNN	28.95	0.859	24.93	0.744	22.13	0.669	563
	SDeCNN	31.57	0.941	26.19	0.823	22.73	0.730	38
	E3DTV	29.69	0.927	31.19	0.943	24.26	0.845	32
	ℓ_0-ℓ_1 HTV	31.78	0.911	<u>34.65</u>	0.949	24.70	0.835	215
	HLRTF*	31.02	0.920	33.73	0.923	<u>24.79</u>	0.832	51
	HLRTF	**35.13**	**0.953**	**35.50**	<u>0.961</u>	**34.42**	**0.951**	58

nonlocal low-rank and sparse factorization (SNLRSF) [5], the residual convolutional neural network (D-CNN) [53], the deep convolutional neural network (SDeCNN) [36], the enhanced 3DTV regularization (E3DTV) [40], and the ℓ_0-ℓ_1 hybrid total variation (ℓ_0-ℓ_1 HTV) [47].

Snapshot Spectral Imaging The snapshot spectral imaging aims at recovering the underlying multi-dimensional image from its low-dimensional measurement and masks [31]. We adopt two HSIs (*WDC mall* and *Pavia*) and three MSIs (*Toy*, *Flowers* [50], and *Bird*[6]) to conduct the experiments. Due to the high computational costs of the compared method DeSCI [31], all datasets are cut with 10 spectral bands. For the same reason, the spatial resolution of MSI *Bird* is manually reduced. The SRs of the masks are set to 0.1, 0.3, and 0.5. The compared methods include the generalized alternating projection (GAP)-based methods (GAPTV [54], GAPBM4D [31], and DeSCI [31]), the shearlet transform and sparsity-based method (SeSCI [49], the plug-and-play deep neural network-based method (PnP) [55], the combined TV and PnP (PnP-TV) [43], and the combined 3DTV and PnP (PnP-3DTV) [43].

4.2.2 Experimental Results

The quantitative results of multi-dimensional image completion, denoising, and snapshot spectral imaging are respectively reported in Tables 2-4. We can observe that the proposed HLRTF* and HLRTF perform competitively

Table 4. The average quantitative results for snapshot spectral imaging. The **best** and <u>second-best</u> values are highlighted.

Sampling rate		0.1		0.3		0.5		Time
Data	Method	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	(s)
HSIs *WDC mall* (256×256×10) *Pavia* (200×200×10)	GAPTV	20.57	0.739	21.81	0.751	22.17	0.742	51
	GAPBM4D	28.43	0.918	27.42	0.902	24.93	0.839	263
	DeSCI	29.04	<u>0.924</u>	28.07	0.910	26.24	0.869	14685
	SeSCI	21.25	0.694	22.03	0.698	22.05	0.672	256
	PnP	17.55	0.560	18.26	0.582	19.17	0.612	239
	PnP-TV	23.32	0.848	24.21	0.842	24.09	0.813	33
	PnP-3DTV	22.80	0.803	25.17	0.830	25.28	0.824	33
	HLRTF*	<u>29.60</u>	<u>0.924</u>	<u>29.93</u>	<u>0.928</u>	<u>28.38</u>	<u>0.905</u>	136
	HLRTF	**30.90**	**0.938**	**30.90**	**0.937**	**30.46**	**0.936**	156
MSIs *Toy* (256×256×10) *Flowers* (256×256×10)	GAPTV	25.35	0.736	26.16	0.771	26.10	0.760	60
	GAPBM4D	29.53	0.852	28.23	0.818	26.60	0.766	324
	DeSCI	30.52	0.860	30.67	0.856	30.04	0.863	16756
	SeSCI	26.49	0.832	26.77	0.846	26.28	0.837	350
	PnP	22.86	0.807	23.16	0.765	23.60	0.742	158
	PnP-TV	28.88	0.898	29.09	0.873	29.29	0.838	39
	PnP-3DTV	31.74	0.914	31.11	0.888	29.58	0.848	42
	HLRTF*	<u>31.85</u>	<u>0.916</u>	<u>31.59</u>	<u>0.911</u>	<u>31.12</u>	<u>0.897</u>	138
	HLRTF	**33.23**	**0.948**	**34.07**	**0.955**	**34.23**	**0.958**	153
MSIs *Bird* (235×341×10)	GAPTV	24.92	0.780	26.31	0.808	26.20	0.792	28
	GAPBM4D	21.73	0.892	21.17	0.806	20.81	0.701	208
	DeSCI	25.28	0.851	26.76	0.874	27.45	0.872	8904
	SeSCI	24.29	0.778	25.80	0.827	25.99	0.839	277
	PnP	21.55	0.788	21.53	0.780	22.17	0.752	96
	PnP-TV	28.30	0.927	28.78	0.915	28.76	0.886	34
	PnP-3DTV	28.10	<u>0.940</u>	<u>30.98</u>	<u>0.936</u>	30.03	0.907	27
	HLRTF*	<u>29.25</u>	0.922	30.64	0.924	<u>30.47</u>	<u>0.916</u>	74
	HLRTF	**30.46**	**0.941**	**31.51**	**0.941**	**32.18**	**0.936**	87

against compared methods, while HLRTF achieves better results than HLRTF* due to the ability of PTV to explore the local smoothness of multi-dimensional images. From the running time comparison, we can observe that the proposed methods are efficient compared with other methods.

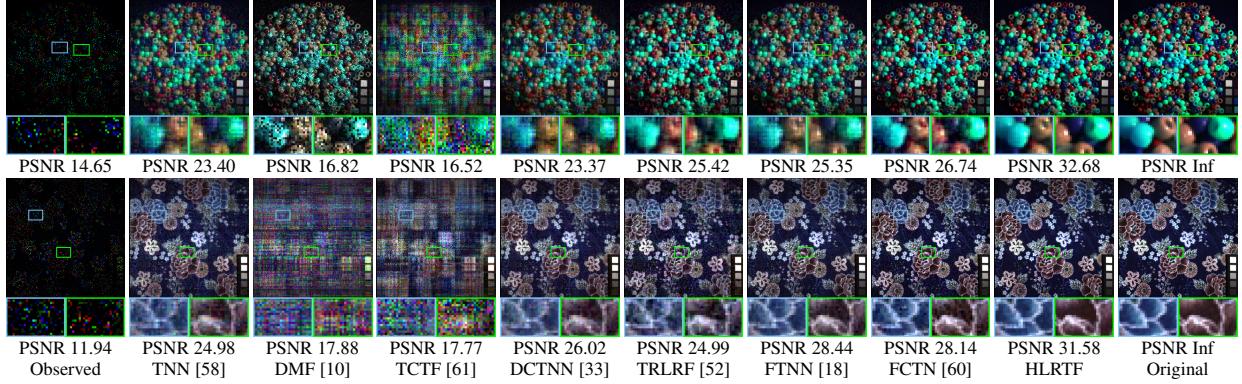Some visual results for different tasks are shown in Figs.

---

[6] https://github.com/liuyang12/DeSCI/tree/master/dataset

Figure 5. The results of multi-dimensional image completion by different methods on MSI *Beans* and MSI *Cloth* (SR = 0.1).

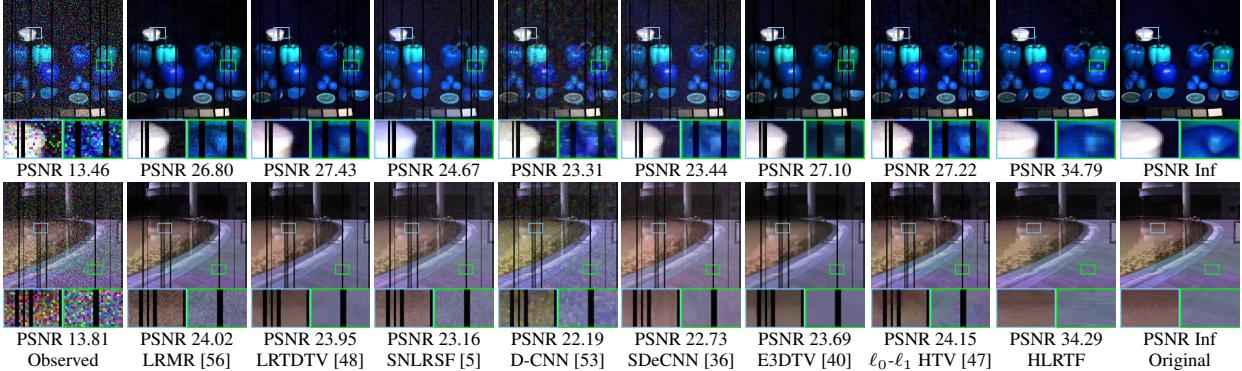| PSNR 14.65 | PSNR 23.40 | PSNR 16.82 | PSNR 16.52 | PSNR 23.37 | PSNR 25.42 | PSNR 25.35 | PSNR 26.74 | PSNR 32.68 | PSNR Inf |
| PSNR 11.94 | PSNR 24.98 | PSNR 17.88 | PSNR 17.77 | PSNR 26.02 | PSNR 24.99 | PSNR 28.44 | PSNR 28.14 | PSNR 31.58 | PSNR Inf |
| Observed | TNN [58] | DMF [10] | TCTF [61] | DCTNN [33] | TRLRF [52] | FTNN [18] | FCTN [60] | HLRTF | Original |

Figure 6. The results of MSI denoising by different methods on MSI *Fruits* and MSI *Pool* for **Case 3**.

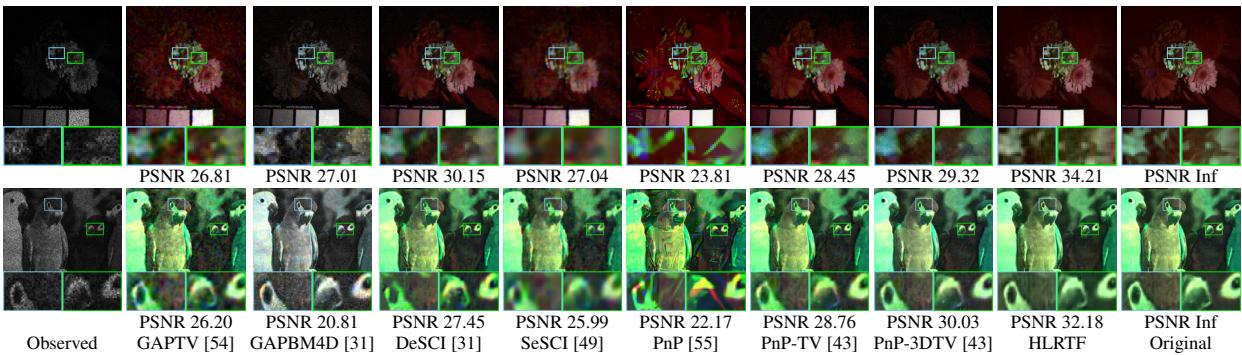| PSNR 13.46 | PSNR 26.80 | PSNR 27.43 | PSNR 24.67 | PSNR 23.31 | PSNR 23.44 | PSNR 27.10 | PSNR 27.22 | PSNR 34.79 | PSNR Inf |
| PSNR 13.81 | PSNR 24.02 | PSNR 23.95 | PSNR 23.16 | PSNR 22.19 | PSNR 22.73 | PSNR 23.69 | PSNR 24.15 | PSNR 34.29 | PSNR Inf |
| Observed | LRMR [56] | LRTDTV [48] | SNLRSF [5] | D-CNN [53] | SDeCNN [36] | E3DTV [40] | ℓ_0-ℓ_1 HTV [47] | HLRTF | Original |

Figure 7. The results of snapshot spectral imaging by different methods on MSI *Flowers* and MSI *Bird* (SR = 0.5).

	PSNR 26.81	PSNR 27.01	PSNR 30.15	PSNR 27.04	PSNR 23.81	PSNR 28.45	PSNR 29.32	PSNR 34.21	PSNR Inf
	PSNR 26.20	PSNR 20.81	PSNR 27.45	PSNR 25.99	PSNR 22.17	PSNR 28.76	PSNR 30.03	PSNR 32.18	PSNR Inf
Observed	GAPTV [54]	GAPBM4D [31]	DeSCI [31]	SeSCI [49]	PnP [55]	PnP-TV [43]	PnP-3DTV [43]	HLRTF	Original

5-7. In Fig. 5, we can observe that HLRTF recovers the multi-dimensional images and captures the fine details better than compared methods with the help of the representation ability brought by the DNN. In Fig. 6, we can observe that HLRTF removes the challenging deadline noise with the help of PTV regularization, while compared methods fail to do so. In Fig. 7, we can observe that HLRTF has cleaner results and can preserve the image details and colors better than compared methods. Please refer to supplementary materials for more visual results.

5. Conclusion

In this paper, we embed a DNN in the t-SVD framework and propose the HLRTF for multi-dimensional image recovery. To address the vanishing gradient issue, we propose the PTV regularization. Extensive experiments on multi-dimensional image completion, denoising, and snapshot spectral imaging verify the effectiveness and superiority of our method over state-of-the-art methods.

References

[1] Animashree Anandkumar, Rong Ge, Daniel Hsu, Sham M. Kakade, and Matus Telgarsky. Tensor decompositions for learning latent variable models. *Journal of Machine Learning Research*, 15(80):2773–2832, 2014.

[2] Sanjeev Arora, Nadav Cohen, Wei Hu, and Yuping Luo. Implicit regularization in deep matrix factorization. In *NeurIPS, 2019*, volume 32, 2019.

[3] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994.

[4] Philip Blagoveschensky and Anh Huy Phan. Deep convolutional tensor network. In *NeurIPS, 2020*, 2020.

[5] Chunhong Cao, Jie Yu, Chengyao Zhou, Kai Hu, Fen Xiao, and Xieping Gao. Hyperspectral image denoising via subspace-based nonlocal low-rank and sparse factorization. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(3):973–988, 2019.

[6] Huiyuan Chen and Jing Li. Neural tensor model for learning multi-aspect factors in recommender systems. In *IJCAI, 2020*, pages 2449–2455, 2020.

[7] Zhengyu Chen, Ziqing Xu, and Donglin Wang. Deep transfer tensor decomposition with orthogonal constraint for recommender systems. In *AAAI, 2021*, pages 4010–4018, 2021.

[8] Miaomiao Cheng, Liping Jing, and Michael K. Ng. A weighted tensor factorization method for low-rank tensor completion. In *IEEE BigMM*, pages 30–38, 2019.

[9] Jen-Tzung Chien and Yi-Ting Bao. Tensor-factorized neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 29(5):1998–2011, 2018.

[10] Jicong Fan and Jieyu Cheng. Matrix completion by deep matrix factorization. *Neural Networks*, 98:34–41, 2018.

[11] Silvia Gandy, Benjamin Recht, and Isao Yamada. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems*, 27(2):025010, 2011.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV, 2015*, pages 1026–1034, 2015.

[13] Christopher J. Hillar and Lek-Heng Lim. Most tensor problems are NP-hard. *Journal of the ACM*, 60(6), 2013.

[14] Huyan Huang, Yipeng Liu, Zhen Long, and Ce Zhu. Robust low-rank tensor ring completion. *IEEE Transactions on Computational Imaging*, 6:1117–1126, 2020.

[15] Masaaki Imaizumi, Takanori Maehara, and Kohei Hayashi. On tensor train rank minimization : Statistical efficiency and scalable algorithm. In *NeurIPS, 2017*, volume 30, 2017.

[16] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456, 2015.

[17] Prateek Jain and S. Oh. Provable tensor factorization with missing data. In *NeurIPS, 2014*, page 1431–1439, 2014.

[18] Tai-Xiang Jiang, Michael K. Ng, Xi-Le Zhao, and Ting-Zhu Huang. Framelet representation of tensor nuclear norm for third-order tensor completion. *IEEE Transactions on Image Processing*, 29:7233–7244, 2020.

[19] Tai-Xiang Jiang, Xi-Le Zhao, Hao Zhang, and Michael K. Ng. Dictionary learning with low-rank coding coefficients for tensor completion. *IEEE Transactions on Neural Networks and Learning Systems*, 2021. doi=10.1109/TNNLS.2021.3104837.

[20] Eric Kernfeld, Misha Kilmer, and Shuchin Aeron. Tensor–tensor products with invertible linear transforms. *Linear Algebra and its Applications*, 485:545–570, 2015.

[21] Misha E. Kilmer, Karen Braman, Ning Hao, and Randy C. Hoover. Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging. *SIAM Journal on Matrix Analysis and Applications*, 34(1):148–172, 2013.

[22] Misha E. Kilmer and Carla D. Martin. Factorization strategies for third-order tensors. *Linear Algebra and its Applications*, 435(3):641–658, 2011.

[23] Diederik Kingma and Jimmy Ba. ADAM: A method for stochastic optimization. In *ICLR, 2014*, 2014.

[24] Tamara G. Kolda and Brett W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.

[25] Hao Kong, Canyi Lu, and Zhouchen Lin. Tensor Q-rank: New data dependent tensor rank. *Machine Learning*, 2021.

[26] Xutao Li, Yunming Ye, and Xiaofei Xu. Low-rank tensor completion with total variation for visual data inpainting. In *AAAI, 2017*, page 2210–2216, 2017.

[27] Hanpeng Liu, Yaguang Li, Michael Tsang, and Yan Liu. Costco: A neural tensor completion model for sparse tensors. In *SIGKDD, 2019*, page 324–334, 2019.

[28] Ji Liu, Przemyslaw Musialski, Peter Wonka, and Jieping Ye. Tensor completion for estimating missing values in visual data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):208–220, 2013.

[29] Na Liu, Wei Li, Ran Tao, and James E. Fowler. Wavelet-domain low-rank/group-sparse destriping for hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 57(12):10310–10321, 2019.

[30] Xiao-Yang Liu, Shuchin Aeron, Vaneet Aggarwal, and Xiaodong Wang. Low-tubal-rank tensor completion using alternating minimization. *IEEE Transactions on Information Theory*, 66(3):1714–1737, 2020.

[31] Yang Liu, Xin Yuan, Jinli Suo, David J. Brady, and Qionghai Dai. Rank minimization for snapshot compressive imaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(12):2990–3006, 2019.

[32] Canyi Lu, Jiashi Feng, Yudong Chen, Wei Liu, Zhouchen Lin, and Shuicheng Yan. Tensor robust principal component analysis with a new tensor nuclear norm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4):925–938, 2020.

[33] Canyi Lu, Xi Peng, and Yunchao Wei. Low-rank tensor completion with a new tensor nuclear norm induced by invertible linear transforms. In *CVPR, 2019*, pages 5989–5997, 2019.

[34] Yi-Si Luo, Xi-Le Zhao, Tai-Xiang Jiang, Yu-Bang Zheng, and Yi Chang. Hyperspectral mixed noise removal via spatial-spectral constrained unsupervised deep image prior. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:9435–9449, 2021.

[35] Baburaj Madathil and Sudhish N. George. DCT based weighted adaptive multi-linear data completion and denoising. *Neurocomputing*, 318:120–136, 2018.

[36] Alessandro Maffei, Juan M. Haut, Mercedes Eugenia Paoletti, Javier Plaza, Lorenzo Bruzzone, and Antonio Plaza. A single model CNN for hyperspectral image denoising. *IEEE Transactions on Geoscience and Remote Sensing*, 58(4):2516–2529, 2020.

[37] Michael K. Ng, Xiongjun Zhang, and Xi-Le Zhao. Patched-tube unitary transform for robust tensor completion. *Pattern Recognition*, 100:107181, 2020.

[38] Alexander Novikov, Dmitry Podoprikhin, Anton Osokin, and Dmitry Vetrov. Tensorizing neural networks. In *NeurIPS, 2015*, page 442–450, 2015.

[39] I. V. Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.

[40] Jiangjun Peng, Qi Xie, Qian Zhao, Yao Wang, Leung Yee, and Deyu Meng. Enhanced 3DTV regularization and its applications on HSI denoising and compressed sensing. *IEEE Transactions on Image Processing*, 29:7889–7903, 2020.

[41] Yi Peng, Deyu Meng, Zongben Xu, Chenqiang Gao, Yi Yang, and Biao Zhang. Decomposable nonlocal tensor dictionary learning for multispectral image denoising. In *CVPR, 2014*, pages 2949–2956, 2014.

[42] Anh-Huy Phan, Konstantin Sobolev, Konstantin Sozykin, Dmitry Ermilov, Julia Gusak, Petr Tichavský, Valeriy Glukhov, Ivan Oseledets, and Andrzej Cichocki. Stable low-rank tensor decomposition for compression of convolutional neural network. In *ECCV, 2020*, pages 522–539, 2020.

[43] Haiquan Qiu, Yao Wang, and Deyu Meng. Effective snapshot compressive-spectral imaging via deep denoising and total variation priors. In *CVPR, 2021*, pages 9127–9136, 2021.

[44] Bernardino Romera-Paredes and Massimiliano Pontil. A new convex relaxation for tensor completion. In *NeurIPS*, page 2967–2975, 2013.

[45] Guangjing Song, Michael K. Ng, and Xiongjun Zhang. Robust tensor completion using transformed tensor singular value decomposition. *Numerical Linear Algebra with Applications*, 27(3):e2299, 2020.

[46] Jian-Li Wang, Ting-Zhu Huang, Xi-Le Zhao, Tai-Xiang Jiang, and Michael K. Ng. Multi-dimensional visual data completion via low-rank tensor representation under coupled transform. *IEEE Transactions on Image Processing*, 30:3581–3596, 2021.

[47] Minghua Wang, Qiang Wang, Jocelyn Chanussot, and Danfeng Hong. ℓ_0-ℓ_1 hybrid total variation regularization and its applications on hyperspectral image mixed noise removal and compressed sensing. *IEEE Transactions on Geoscience and Remote Sensing*, 2021.

[48] Yao Wang, Jiangjun Peng, Qian Zhao, Yee Leung, Xi-Le Zhao, and Deyu Meng. Hyperspectral image restoration via total variation regularized low-rank tensor decomposition.

IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 11(4):1227–1243, 2018.

[49] Peihao Yang, Linghe Kong, Xiao-Yang Liu, Xin Yuan, and Guihai Chen. Shearlet enhanced snapshot compressive imaging. *IEEE Transactions on Image Processing*, 29:6466–6481, 2020.

[50] Fumihito Yasuma, Tomoo Mitsunaga, Daisuke Iso, and Shree K. Nayar. Generalized assorted pixel camera: Post-capture control of resolution, dynamic range, and spectrum. *IEEE Transactions on Image Processing*, 19(9):2241–2253, 2010.

[51] Fanghua Ye, Chuan Chen, and Zibin Zheng. Deep autoencoder-like nonnegative matrix factorization for community detection. In *CIKM, 2018*, page 1393–1402, 2018.

[52] Longhao Yuan, Chao Li, Danilo P. Mandic, Jianting Cao, and Qibin Zhao. Tensor ring decomposition with rank minimization on latent space: An efficient approach for tensor completion. In *AAAI, 2019*, pages 9151–9158, 2019.

[53] Qiangqiang Yuan, Qiang Zhang, Jie Li, Huanfeng Shen, and Liangpei Zhang. Hyperspectral image denoising employing a spatial–spectral deep residual convolutional neural network. *IEEE Transactions on Geoscience and Remote Sensing*, 57(2):1205–1218, 2019.

[54] Xin Yuan. Generalized alternating projection based total variation minimization for compressive sensing. In *ICIP, 2016*, pages 2539–2543, 2016.

[55] Xin Yuan, Yang Liu, Jinli Suo, and Qionghai Dai. Plug-and-play algorithms for large-scale snapshot compressive imaging. In *CVPR, 2020*, pages 1447 – 1457, 2020.

[56] Hongyan Zhang, Wei He, Liangpei Zhang, Huanfeng Shen, and Qiangqiang Yuan. Hyperspectral image restoration using low-rank matrix recovery. *IEEE Transactions on Geoscience and Remote Sensing*, 52(8):4729–4743, 2014.

[57] Shipeng Zhang, Lizhi Wang, Lei Zhang, and Hua Huang. Learning tensor low-rank prior for hyperspectral image reconstruction. In *CVPR, 2021*, pages 12006–12015, 2021.

[58] Zemin Zhang, Gregory Ely, Shuchin Aeron, Ning Hao, and Misha Kilmer. Novel methods for multilinear data completion and de-noising based on tensor-svd. In *CVPR, 2014*, pages 3842–3849, 2014.

[59] Yu-Bang Zheng, Ting-Zhu Huang, Teng-Yu Ji, Xi-Le Zhao, Tai-Xiang Jiang, and Tian-Hui Ma. Low-rank tensor completion via smooth matrix factorization. *Applied Mathematical Modelling*, 70:677–695, 2019.

[60] Yu-Bang Zheng, Ting-Zhu Huang, Xi-Le Zhao, Qibin Zhao, and Tai-Xiang Jiang. Fully-connected tensor network decomposition and its application to higher-order tensor completion. In *AAAI, 2021*, 2021.

[61] Pan Zhou, Canyi Lu, Zhouchen Lin, and Chao Zhang. Tensor factorization for low-rank tensor completion. *IEEE Transactions on Image Processing*, 27(3):1152–1163, 2018.