Pattern Recognition 172 (2026) 112612

Contents lists available at ScienceDirect

Pattern Recognition

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

Degradation accordant plug-and-play for low-rank tensor recovery

Yexun Hu [a,b,c], Zixin Tang [a,b,c], Tai-Xiang Jiang [a,b,c,*], Xi-Le Zhao [d], Guisong Liu [a,b,c]

[a] School of Computing and Artificial Intelligence, Southwestern University of Finance and Economics, Chengdu, 611130, China
[b] Kash Institute of Electronics and Information Industry, Kash, 844000, China
[c] Engineering Research Center of Intelligent Finance, Ministry of Education, Southwestern University of Finance and Economics, Chengdu, 611130, China
[d] School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, 611731, China

ARTICLE INFO

Keywords:
Noisy tensor completion
Plug-and-play framework
Tensor nuclear norm
Low-rank coding

ABSTRACT

This work aims to recover multi-dimensional visual data from limited partial sampling with noisy observations. For this typical inverse problem, we first employ the discrete cosine transform (DCT)-based tensor nuclear norm (TNN) to depict the global low-rank structure. Then, the overall degradation process is decoupled into two parts, i.e., the entries missing and the noise corruption. To recover abundant details in visual data effectively, we employ the plug-and-play (PnP) prior framework. Specifically, we formulate two implicit regularization terms to integrate two deep convolutional neural networks (CNNs), which are pre-trained for image denoising and completion, respectively. The proposed model, which consists of the low-rank part and PnP modules, is efficiently solved using the alternating direction method of multipliers (ADMM). and the theoretical guarantee of its convergence is then established. Numerical experiments are conducted to demonstrate the superiority of the proposed method compared to state-of-the-art methods on various types of multi-dimensional visual data, and the excellent performance on compressive sensing magnetic resonance imaging (CS-MRI) validates the promising generalization ability for different multi-dimensional image recovery tasks of the proposed model.

1. Introduction

With the widespread use of multi-sensor technology, many real-world visual data, e.g., videos [1], hyper/multi-spectral images (HSIs/MSIs) [2], and magnetic resonance imaging (MRI) data [3], are often multi-dimensional. However, limitations of imaging environments or devices for real-world applications often result in image degradation, such as noise corruption, blur, and under-sampling, which can seriously affect subsequent applications like semantic segmentation and object detection. Therefore, it is essential to recover high-quality multi-dimensional imaging data from degraded observations. In this study, we focus on a typical multi-dimensional visual data recovery problem, namely the noisy tensor completion, where the partially observed entries of original visual data are corrupted with noise. When the degradation is determined, introducing reasonable prior knowledge of the visual data within the maximum a posteriori (MAP) framework becomes the critical point. On account of the low-dimensional structures maintained by many real-world multi-dimensional visual data, researchers have made significant efforts to exploit the matrix/tensor low-rankness to characterize the overall low-dimensionality [4]. Unlike the matrix case, the tensor format (i.e., the multi-way array) is naturally suitable to represent the multi-dimensional visual data, preserving their inherent structural information better. Therefore, tensor low-rankness is widely utilized for restoring multi-dimensional imaging data and has achieved great success in the past decade [5].

However, the rank of the tensor is not uniquely defined, such as the Tucker rank [6,7], the CANDECOMP/PARAFAC (CP) rank [8], the tensor tubal-rank [9,10], which are derived from various tensor decomposition paradigms. Among these definitions of tensor rank, the last one, defined based on the tensor singular value decomposition (t-SVD) [9] framework which is originally defined specifically for third-order tensors, is fit for processing various multidimensional imaging data. As minimizing the tubal-rank is NP-hard, the tensor tubal nuclear norm (TNN), a convex surrogate of the tensor tubal-rank, and the theoretical guarantee for exact recovery are proposed in [10]. Consequently, the t-SVD has garnered considerable attention in recent studies and shown results in different tensor recovery tasks [11–13].

Meanwhile, multidimensional images retain abundant spatial details, which could not be fully captured by the tensor low-rankness prior alone. Therefore, some additional prior knowledge is necessary to better preserve these details. For example, the nonlocal regularizer is introduced in [15] to characterize the nonlocal self-similarity. Alternatively,

* Corresponding author.
E-mail addresses: huyexun2022@163.com (Y. Hu), sctangzixing123@163.com (Z. Tang), taixiangjiang@gmail.com (T.-X. Jiang), xlzhao122003@163.com (X.-L. Zhao), gliu@swufe.edu.cn (G. Liu).

https://doi.org/10.1016/j.patcog.2025.112612
Received 25 February 2025; Received in revised form 11 October 2025; Accepted 13 October 2025
Available online 17 October 2025

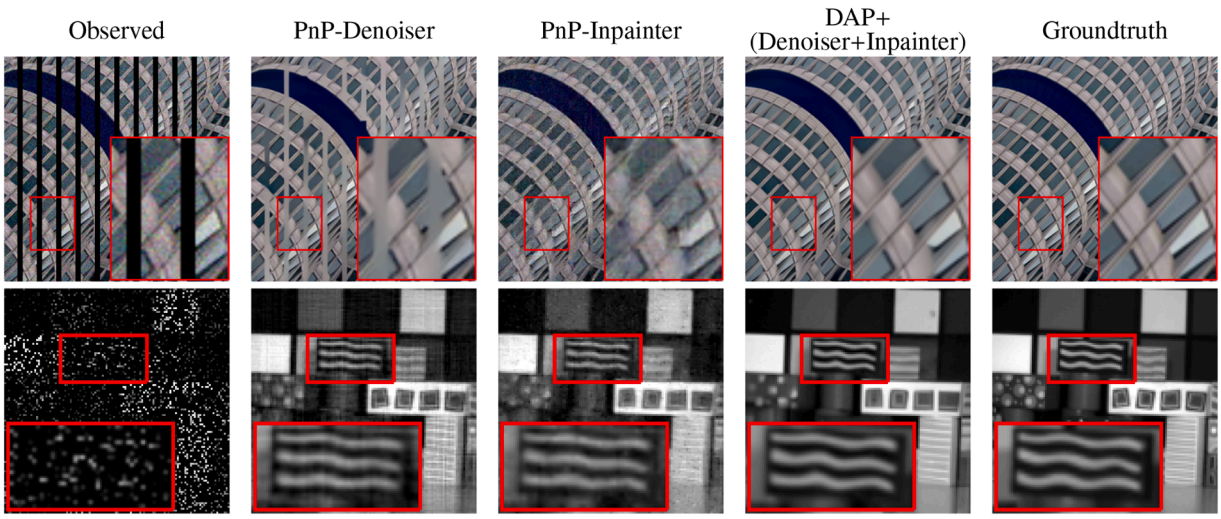

Fig. 1. The recovered results from different PnP-based frameworks on the color image (#68 in Urban100 [14] dataset) with structural missing (the noisy level $\sigma = 15$) and the HSV data with random missing (SR $= 0.2$, $\sigma = 20$). The PnP-Denoiser struggles with the local structural patterns while the PnP-Inpainter fails to reconstruct the fine details, and the proposed DAP+, which incorporates decoupled degradation information, preserves spatial geometries and details more effectively.

with the help of the plug-and-play (PnP) framework, prior knowledge from a denoising algorithm can be incorporated [16]. Moreover, as deep learning advances rapidly, various deep learning-based denoising techniques, which are pretrained on plenty of natural images, can be adopted for leveraging the data-driven prior knowledge within the flexible PnP framework, and the promising performance of this approach is demonstrated by the excellent experimental results of [17,18]. Nevertheless, [19] notes that the plugged-in deep learning-based algorithms, with a high model capacity, are not limited to expressing data-driven priors. They are also expected to undertake part of the degradation, and better performance can be achieved when the considered degradation (in their training stage) is consistent with that of the problem to be solved.

In this work, we tackle the challenging noisy tensor completion problem, in which the degradation consists of both undersampling and noise corruption. Compared to the counterparts for the tensor completion task, existing methods for noisy tensor completion commonly incorporate a fidelity term $\|\mathcal{P}_\Omega(\mathcal{X}) - \mathcal{O}\|_F^2$ to mitigate the impact of the noise corruption [20]. Although leveraging tensor low-rank priors is proficient at capturing the overall low-dimensional structure of multidimensional visual data, it is still struggling to precisely preserve fine details. From this perspective, the idea of employing pretrained deep neural networks under the PnP framework for well detail preservation in tensor completion methods [17,19] can be considered for noisy tensor completion. However, existing tensor completion methods either consider the deep denoiser [17,21] or the deep inpainter [19] in an independent usage context. The degradation considered in these employed pretrained deep neural networks is not consistent with that of noisy tensor completion, limiting the performance of the direct use of the idea of deep PnP. To address these limitations, we propose a novel approach that explicitly decouples the degradation of noisy tensor completion into two components: pixel loss and noise corruption. Then, two implicit regularizers, whose subproblems are respectively handled by a deep inpainter and a deep denoiser, are established to introduce the degradation accordant data-driven prior.

As discussed above, we propose to combine the tensor low-rank prior with the degradation accordant data-driven prior under the PnP framework. Specifically, we leverage the discrete-cosine-transform (DCT)-based TNN to capture the tensor low-rankness instead of TNN as DCT has been shown to be more suitable for visual data with its reflective boundary [11]. On the other hand, as for the PnP prior, commonly used deep learning methods are typically designed only for either image inpainting (deep inpainter) or denoising (deep denoiser), focusing on reconstructing low-frequency information (e.g., local spatial structures) or high-frequency details (e.g., sharp edges and fine textures), respectively. Therefore, we propose decoupling the degradation into the pixel loss and the noise corruption, and simultaneously introduce two deep convolution neural networks (CNNs), which are pre-trained respectively for denoising and completion, to capture the low-frequency and high-frequency features concurrently. Fig. 1 presents representative examples of a color image (#68 from the Urban100 [14] dataset) and hyperspectral video (HSV) data. Moreover, with the usage of 2D fast Fourier transform (2D FFT), we convert these results on color image #68 into the k-space. In the context of k-space, low-frequency information is predominantly concentrated in the central region, while high-frequency information is distributed across the peripheral areas. Fig. 2 illustrates the error map of the reconstructed results compared to the groundtruth in the Fourier domain on the color image #68. It is evident that our decoupling strategy is very effective not only in preserving low-frequency information but also in mitigating high-frequency distortions, which leads to an excellent performance of the proposed method in depicting fine details and structural patterns in multi-dimensional imaging data.

In a nutshell, our contributions are summarised as follows:

- We propose a novel noisy tensor completion model for multi-dimensional visual data, including videos, multi-spectral images, and hyperspectral videos. In the proposed model, DCTNN is adopted as a low-rank regularizer to exploit the intrinsic low-dimensional structure of tensors. Meanwhile, two implicit regularizers are incorporated to leverage the degradation accordant data-driven prior for effective details and geometric structure preservation within the plug-and-play framework.
- To optimize the proposed model, we design an ADMM-based algorithm. The degradation in noisy tensor completion is decoupled into the noise and the undersampling parts. Thus, our ADMM-based algorithm formulates the two implicit-regularizer-related subproblems as a denoising problem and an inpainting problem, respectively. This allows us to plug in off-the-shelf denoising and inpainting CNNs, which are pretrained on numerous natural image data and are readily accessible. Moreover, we theoretically establish the convergence of the proposed algorithm, which is quite difficult when the implicit regularization and deep neural network are evolved.
- Numerous experiments are conducted on various multi-dimensional visual data. Results demonstrate that our method outperforms state-of-the-art methods, the effectiveness of our degradation decoupling

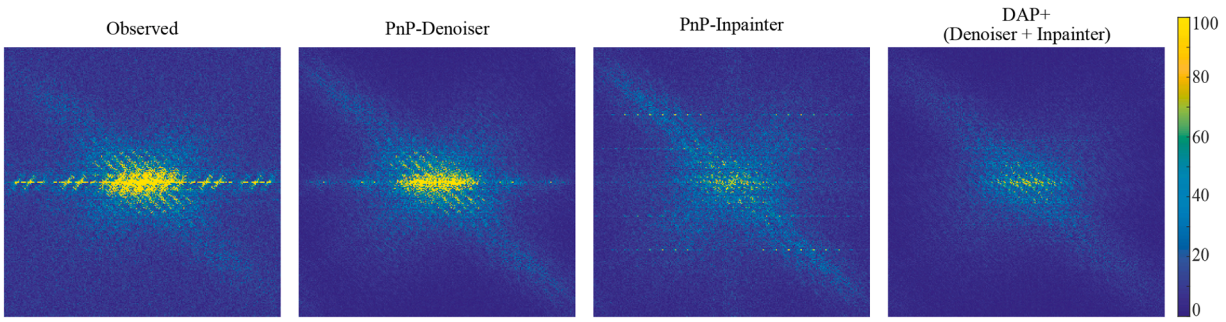

Fig. 2. The amplitude error map of the recovered results (color image #68) in the Fourier domain. The denoiser concentrates more on image details yielding less error in the high-frequency part, while the inpainter can better reconstruct the structure resulting in less error in the low-frequency part. Our DAP+ can take both advantages.

strategy and different low-rankness regularizers are illustrated in the ablation study. Moreover, experiments on compressive sensing MRI show that the proposed method maintains a good generalization ability.

This work is an extension of our previous work published in [19]. The new material is the following: a) the proposed method focuses on a more challenging tensor restoration problem, namely noisy tensor completion; b) we utilize discrete cosine transform (DCT)-based tensor nuclear norm to characterize the global low-rank structure; c) to introduce the data-driven prior and fully leverage the degradation information, a deep denoiser and a deep inpainter are employed as the implicit regularizers within the PnP framework; d) the theoretical guarantee of the convergence of the solving algorithm is established; and e) we present more extensive experimental results on various multi-dimensional images.

The remaining sections of this paper are organized as such: Section 2 presents a brief survey of related work. The fundamental preliminaries related to the tensor are introduced in Section 3. Sections 4 and 5 outline the primary findings of this work. Numeric experiments, ablation studies, and corresponding results are presented in Section 6. Finally, conclusions are drawn in Section 7.

2. Related work

2.1. Low-rank tensor recovery

As mentioned earlier, the definition of tensor rank is not unique, and many types of tensor rank, defined based on different tensor decomposition methods, have been proposed. Tensor decomposition methods typically factorize the tensor into a series of sub-factors or sub-tensors, similar to matrix factorization. and four major decomposition schemes have been widely investigated recently and applied to various tensor recovery tasks, such as tensor completion. One of the earliest and most widely used tensor decomposition techniques is the Tucker decomposition [6]. It decomposes a N-order tensor $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \cdots \times n_N}$ into N submatrices along each mode and a core tensor that characterizes mutual interactions across different modes, which can be formulated as:

$$\mathcal{X} = \mathcal{G} \times_1 \mathbf{A}_1 \times_2 \mathbf{A}_2 \cdots \times_N \mathbf{A}_N,$$

where $\times_k$ denotes the mode-k tensor-matrix multiplication [22], $\mathcal{G} \in \mathbb{R}^{r_1 \times \cdots \times r_N}$ is the core tensor, $\mathbf{A}_k \in \mathbb{R}^{n_k \times r_k}$ $(k = 1, 2, \cdots, N)$ represent the factor matrix, and the N-tuple $(r_1, \ldots, r_N)$ is the Tucker rank of $\mathcal{T}$. The flexibility of capturing mode-wise interactions of tensors makes Tucker decomposition a significant tool. Liu et al. [23] proposed a low-rank tensor completion (LRTC) method, which utilizes the sum of nuclear norms for all mode-n unfolding matrices of a tensor as a convex surrogate of Tucker rank. Tong et al. [24] proposed and validated a formulation that reinterprets the Tucker model as a CP decomposition with low-rank factor matrices, enabling a more direct and effective approach to multilinear rank learning.

The CANDECOMP/PARAFAC (CP) decomposition [8] is a simple and interpretable tensor decomposition technique, which decomposes a tensor into a set of factor vectors along different dimensions and represents the tensor as the sum of rank-one components formed by the outer product of these vectors. The CP decomposition of a tensor $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \cdots \times n_N}$ can be expressed as

$$\mathcal{T} = \sum_{z=1}^{Z} \lambda_z \mathbf{a}^{z,1} \circ \mathbf{a}^{z,2} \circ \cdots \mathbf{a}^{z,N}. \tag{1}$$

where $Z \in \mathbb{N}^+$ is the number of rank-one components, $\lambda_z \in \mathbb{R}$s are scalar weights, $\circ$ denotes the outer product of vectors, and $\mathbf{a}^{z,i} \in \mathbb{R}^{n_i}$. The minimum Z required to express $\mathcal{X}$ is called the CP rank. Although determining the appropriate CP rank for a tensor remains a challenge [25], for many real-world datasets with strong intrinsic low-dimensional structures, a small CP rank often suffices to achieve an effective representation and shows promising performance on various tensor recovery tasks [5].

The tensor network decomposition [26,27] is also an effective way to depict the intrinsic structure of tensors. It provides a framework for factorizing higher-order tensors into a set of interconnected small-scale factor tensors. This significantly reduces computational complexity and memory costs, making it particularly suitable for large-scale tensor data. Representative tensor network decomposition methods include tensor train (TT) decomposition [26] and tensor ring (TR) decomposition [2]. Zheng et al. [28] proposed the fully connected tensor network (FCTN) decomposition to establish the comprehensive correlations between arbitrary two factors, which performs well on various tensor recovery tasks like the traffic data completion.

The tensor singular value decomposition (t-SVD), induced by the well-defined tensor-tensor product (t-prod, see Definition 1), extends the matrix SVD to third-order tensors [9]. For a third-order tensor $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the t-SVD can be formulated as:

$$\mathcal{T} = \mathcal{U} *_t \mathcal{S} *_t \mathcal{V}^H,$$

where $\mathcal{U} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$ and $\mathcal{V} \in \mathbb{R}^{n_2 \times n_2 \times n_3}$ are orthogonal tensors (see Definition 4), $\mathcal{S} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is an f-diagonal tensor (only diagnoal entries of all frontal slices are non-zero), the $\mathcal{V}^H$ is the conjugate transpose of $\mathcal{V}$ (see Definition 2), $*_t$ denotes the t-prod. The tensor tubal rank [29] of $\mathcal{T}$ is defined as the number of non-zero singular tubes (the vectors along the third mode) of $\mathcal{S}$. On account of the excellent power of capturing the global low-dimensional structure information and the "spatial-shifting" correlation in tensors [4], the t-SVD framework has attracted increasing attention from many researchers across different applications. Since minimizing the tensor tubal rank is NP-hard, a convex surrogate in the Fourier domain of tubal rank called the heuristic tensor nuclear norm (TNN) is utilized in the tensor completion in [10] with a theoretical guarantee. Kernfeld et al. [30] showed that t-prod [9] can be defined

based on any invertible linear transforms, which motivated many works introducing various transforms into the t-SVD framework like the discrete cosine transform [11], the tight wavelet frame (framelet) [31]. Additionally, the t-SVD framework has been generalized to handle higher-order tensors, further expanding its applicability in data analysis [32]. Wang et al. [4] build a novel regularizer term that incorporated low-rankness and smoothness together to design a high-order tensor recovery model with theoretical exact-recovery guarantees, which demonstrated impressive performances in tensor completion and tensor robust principal component analysis.

Different tensor decompositions maintain specific advantages and disadvantages, which are briefly summarized as follows. Tucker decomposition offers a flexible approach to dimensionality reduction by factoring a tensor into a smaller core tensor and some factor matrices, and it excels at capturing multilinear relationships within data. However, Tucker decomposition lacks unique solutions [22]. In contrast, the uniqueness and identifiability of the CP decomposition make it ideal for revealing distinct underlying patterns in data, whereas the determination of CP-rank is NP-hard [25]. Tensor Network decompositions are well-suited for handling high-order tensors by factorizing them into small-scaled interconnected factor tensors, facilitating efficient computation and scalability [26]. Nevertheless, existing methods generally rely on heuristic strategies [28] to determine the specific network structure, i.e., how the factor tensors are connected. The t-SVD is analogous to the matrix SVD and is effective at capturing global low-dimensional structures in multidimensional visual data. While classic t-SVD-based tensor decomposition methods were tailored for third-order tensors, some recent advancements, such as [4], generalize to high-order cases. However, the flexibility for handling the heterogeneous correlations along different modes is limited owing to its specific decomposition scheme.

2.2. Plug-and-play based tensor recovery

Although low-rank tensor recovery methods have succeeded substantially in various real-world applications, the recovery of ample details retained in multidimensional images remains a bottleneck. Many researchers introduced additional priors, such as local smoothness and non-local similarity, to improve the effectiveness of recovery. Meanwhile, with the help of the flexible Plug-and-Play (PnP) framework, state-of-the-art denoisers can be utilized in ADMM or other proximal algorithms by formulating an implicit regularizer term [16], and the corresponding subproblem can be regarded as a denoiser problem. However, the limitation of hand-crafted prior on characterizing the data feature is inevitable. Recently, motivated by the success of deep learning, many efforts have been devoted to incorporating deep neural network techniques as a data-driven prior term to develop PnP-based models.

Zhang et al. [18] trained a set of specific denoising convolutional networks and integrated them into a model-based optimization method to solve image deblurring, image denoising, and image super-resolution. Zhao et al. [17] introduced the deep data-driven prior into the TNN-based tensor recovery model and achieved impressive performance in tensor completion and image demosaicing. Hu et al. [19] incorporated degradation information in the tensor completion task and employed deep inpainting networks to introduce degradation accordant deep plug-and-play prior to the LRTC model.

2.3. Degradation decoupling strategy

To the best of our knowledge, there are a limited number of investigations that leverage the idea of degradation decoupling. Liu et al. [33] proposed a video restoration network specifically designed for Under-Display Camera (UDC) systems, where the display's pixel array causes complex degradations like diffraction and light attenuation, leading to varying light intensity and diverse visual artifacts (e.g., flare and haze) over time. The core innovation lies in its decoupling attention module, which utilizes a soft mask generation function to analyze each video frame and decomposes its degradation into distinct flare and haze components, based on the principles of how incident light of varying intensities causes diffraction. Xue et al. [34] focused on the underwater image enhancement task and presented a deep neural network with a parallel architecture, consisting of a model-inspired haze-removal module and a data-driven color-adaptation module, to address the complex degradation caused by underwater environments. These two methods have shown promising performance for their respective tasks and have also demonstrated the effectiveness of the degradation decoupling strategy. It should be noted that these two works leverage the degradation decoupling strategy implicitly through the design of deep network modules, and by utilizing a large amount of labeled datasets to train their end-to-end models. This contrasts with our approach. To the best of our knowledge, our method is the first to explicitly decouple the degradation by the formulation of an optimization model within the plug-and-play framework.

3. Notations and preliminaries

For brevity, we give the main notations for various basic concepts of the tensors in Table 1. Then, we provide the basic definitions necessary for the proposed model.

Definition 1 (Tensor-tensor product [9]). For third-order tensors $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{B} \in \mathbb{R}^{n_2 \times n_4 \times n_3}$, the tensor-tensor product $\mathcal{T} = \mathcal{A} * \mathcal{B}$ is a $n_1 \times n_4 \times n_3$ tensor whose (i_1, i_2)-th tube $\mathcal{T}(i_1, i_2, :)$ is obtained by

$$\mathcal{T}(i_1, i_2, :) = \sum_{l}^{n2} \mathcal{B}(i_1, l, :) \circledast \mathcal{A}(l, i_2, :) \tag{2}$$

where $\circledast$ denotes the circular convolution between two vectors of the same size.

Definition 2 (Conjugate transpose [29]). For third-order tensors $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, its conjugate transpose of $\mathcal{T}$, denotes as $\mathcal{T}^H$, can be obtained by conjugate transposing all the frontal slices of $\mathcal{T}$ and then reversing these slices from 2 to the end except for the first slice.

Definition 3 (Identity tensor [29]). The identity tensor $\mathcal{I} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$ is a tensor whose first frontal slice is an identity matrix and the others are zeros matrices.

Definition 4 (Orthogonal tensor [29]). Given a tensor $\mathcal{T} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$, it is orthogonal if $\mathcal{T} * \mathcal{T}^H = \mathcal{T}^H * \mathcal{T} = \mathcal{I}$.

Theorem 1 (t-SVD [29]). *Given a third-order tensors $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the t-SVD of it can be formulated as*

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

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, $\mathcal{S} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is an f-diagonal tensor, the $\mathcal{V}^H$ is the conjugate transpose of $\mathcal{V}$.

Definition 5 (Block diagonal operation [10]). The block diagonal operation of $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is given by

$$\text{bd}iag(\mathcal{T}) \triangleq \begin{bmatrix} \mathcal{T}^{(1)} & & & \\ & \mathcal{T}^{(2)} & & \\ & & \ddots & \\ & & & \mathcal{T}^{(n_3)} \end{bmatrix}, \tag{4}$$

where $\text{bdiag}(\mathcal{T}) \in \mathbb{C}^{n_1 n_3 \times n_2 n_3}$.

Definition 6 (Tubal nuclear norm (TNN) [10]). The tubal nuclear norm of a third order tensor $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, denoted as $\|\mathcal{T}\|_{\text{TNN}}$, is defined as

$$\|\mathcal{T}\|_{\text{TNN}} \triangleq \sum_{i}^{n_3} \|\widehat{\mathcal{T}}(:, :, i)\|_*, \tag{5}$$

where $\widehat{\mathcal{T}}$ is the result of applying fast Fourier transform (FFT) along the third dimension of $\mathcal{T}$, and $\|\cdot\|_*$ is the matrix nuclear norm, i.e., the sum of singular values.

Table 1
Summary of notations.

Symbol	Description	Symbol	Description
$s \in \mathbb{R}$ $\mathbf{M} \in \mathbb{R}^{m\times n}$	Scalar Matrix	$\mathbf{v} \in \mathbb{R}^{n}$ $\mathcal{T} \in \mathbb{R}^{n_1\times\cdots\times n_N}$	Vector Tensor
$\mathcal{T}_{i_1 i_2\cdots i_N}$ or $\mathcal{T}(i_1, i_2, \ldots, i_N)$	Tensor element	$\mathcal{T}(i, j, :)$	The (i, j)-th tube of a third-order tensor $\mathcal{T}$
$\mathcal{T}(:, :, k)$ or $\mathcal{T}_{(k)}$	The k-th frontal slice of a third-order tensor $\mathcal{T}$	$\langle \mathcal{T}, \mathcal{S}\rangle =$ $\sum_{i_1,\ldots,i_N} \mathcal{T}_{i_1\cdots i_N} \mathcal{S}_{i_1\cdots i_N}$	Tensor inner product
$\|\mathcal{T}\|_F =$ $\sqrt{\langle \mathcal{T}, \mathcal{T}\rangle}$	Tensor Frobenius norm	$\mathcal{A} * \mathcal{B}$	The tensor-tensor product (t-prod) of two third-order tensors $\mathcal{A}$ and $\mathcal{B}$
vec$(\cdot)$	The vectorization of a tensor $\mathbb{R}^{n_1\times n_2\times n_3} \to \mathbb{R}^{n_1 n_2 n_3\times 1}$	vec$^{-1}(\cdot)$	The inverse operation of vec$(\cdot)$ $\mathbb{R}^{n_1 n_2 n_3\times 1} \to \mathbb{R}^{n_1\times n_2\times n_3}$

It is noteworthy that the FFT utilized in TNN could be replaced by other linear invertible transforms, e.g., the DCT is adopted in [11] to take the place of the FFT, i.e., $\|\mathcal{T}\|_\star \triangleq \sum_{i=1}^{n_3} \|\tilde{\mathcal{T}}(:, :, i)\|_*$, where $\|\mathcal{T}\|_\star$ denotes the DCT-based TNN and $\tilde{\mathcal{T}}$ is the DCT transformed tensor (along the third mode) of $\mathcal{T}$.

4. The proposed model and algorithm

The general plug-and-play based low-rank tensor recovery (LRTR) model could be formulated as follows:

$$\min_{\mathcal{X}} \Phi(\mathcal{X}) + \Psi(\mathcal{X}). \text{ s.t. } \mathcal{A}(\mathcal{X}) + \mathcal{N} = \mathcal{O}. \tag{6}$$

where $\mathcal{O} \in \mathbb{R}^{n_1\times\cdots\times n_L}$ is the measurement tensor, L is the dimension amount, $\mathcal{X} \in \mathbb{R}^{n_1\times\cdots\times n_L}$ is the underlying tensor, $\mathcal{N}$ represents the dense additive noise (we mainly consider the zero-mean Gaussian noise in this work), $\mathcal{A}(\cdot)$ is the linear forward operator (in tensor completion, it refers to a projection operator $\mathcal{P}_\Omega(\cdot)$ which maintains the values of the elements in a specific set Ω while assigning zero to other elements), $\Phi(\mathcal{X})$ denotes a general regularizer of tensor low-rankness, and $\Psi(\mathcal{X})$ is an implicit regularizer to introduce the PnP prior. In this work, we leverage the DCT-based TNN [11] to capture the intrinsic low-dimensional structures of multidimensional data. Meanwhile, on account of the zero-mean Gaussian noise we thinking about, (6) can be reformulated as

$$\min_{\mathcal{X}} \lambda\|\mathcal{X}\|_\star + \Psi(\mathcal{X}) + \mu\Big(\frac{1}{2}\|\mathcal{A}(\mathcal{X}) - \mathcal{O}\|_F^2\Big) + \frac{\rho}{2}\|\mathcal{X}\|_F^2, \tag{7}$$

where $\|\mathcal{X}\|_\star$ denotes the DCT-based TNN of $\mathcal{X}$, $\|\mathcal{X}\|_F^2$ is a Tikhonov reuglarizer, λ, μ and ρ are nonnegative trade-off parameters.

Then, we split $\Psi(\mathcal{X})$ into two implicit regularizers: $\psi_1(\mathcal{X})$ and $\psi_2(\mathcal{X})$. We can see in the following part that this simple step enables us to decouple the hybrid degradation clearly into the denoising part and the inpainting part reasonably. Thus, more pretrained CNNs customized for denoising or inpainting are readily available as finding a suitable CNN pretrained for both tasks simultaneously is not that easy. Meanwhile, the plugged-in CNNs also obey the concept of degradation accordant in [19]. Next, after introducing auxiliary variables, the optimization problem turns out to be

$$\begin{aligned}\min_{\mathcal{X},\mathcal{Y},\mathcal{Z},\mathcal{W}}\ & \lambda\|\mathcal{W}\|_\star + \psi_1(\mathcal{Y}) + \psi_2(\mathcal{Z}) + \mu\Big(\frac{1}{2}\|\mathcal{A}(\mathcal{X}) - \mathcal{O}\|_F^2 + \frac{\rho}{2}\|\mathcal{X}\|_F^2\Big)\\ &\text{s.t. } \mathcal{A}(\mathcal{Z}) = \mathcal{A}(\mathcal{X}),\ \mathcal{Y} = \mathcal{X},\ \mathcal{W} = \mathcal{X},\end{aligned} \tag{8}$$

where $\mathcal{W}$, $\mathcal{Z}$, and $\mathcal{Y} \in \mathbb{R}^{n_1\times n_2\cdots\times n_L}$ are auxiliary variables. We remark that the constraint $\mathcal{A}(\mathcal{Z}) = \mathcal{A}(\mathcal{X})$ requires the auxiliary variable $\mathcal{Z}$ admits to $\mathcal{X}$ after applying the linear forward operator. Therefore, (8) is not strictly equivalent to (7). The benefit of introducing the auxiliary variable in this usual way is that we can introduce the inpainting networks (See the $\mathcal{Z}$ related subproblem).

Thus, the augmented Lagrangian function of (8) could be written as follow:

$$\begin{aligned}L(\mathcal{X},\mathcal{W},\mathcal{Z},\Lambda) =& \lambda\|\mathcal{W}\|_\star + \beta\|\mathcal{W} - \mathcal{X} + \frac{\Lambda_1}{\beta}\|_F^2 + \frac{\beta}{2}\|\mathcal{Y} - \mathcal{X} + \frac{\Lambda_2}{\beta}\|_F^2\\ &+ \frac{\beta}{2}\|\mathcal{A}(\mathcal{Z}) - \mathcal{A}(\mathcal{X}) + \frac{\Lambda_3}{\beta}\|_F^2 + \psi_1(\mathcal{Y}) + \psi_2(\mathcal{Z})\\ &+ \mu\Big(\frac{1}{2}\|\mathcal{A}(\mathcal{X}) - \mathcal{O}\|_F^2 + \frac{\rho}{2}\|\mathcal{X}\|_F^2\Big),\end{aligned} \tag{9}$$

where β is the nonnegative penalty parameter, Λ_1, Λ_2, and Λ_3 are the multipliers.

Then, following standard steps of the ADMM [35], each variable is updated by iteratively solving the corresponding subproblem.

1) The $\mathcal{W}$ subproblem is

$$\min_{\mathcal{W}} \lambda\|\mathcal{W}\|_\star + \beta\|\mathcal{W} - \mathcal{X}^k + \frac{\Lambda_1^k}{\beta}\|_F^2, \tag{10}$$

where k is the number of the iteration, and by the application of tensor singular value thresholding (t-SVT) [25], the closed-form solution of each $\mathcal{W}$ could be exactly determined as

$$\mathcal{W}^{k+1} = \text{t-SVT}_{\frac{\lambda}{\beta}}\Big(\mathcal{X}^k - \frac{\Lambda_1}{\beta}\Big) \triangleq \mathcal{U}\Big(\mathcal{S} - \frac{\lambda}{\beta}\Big)\mathcal{V}, \tag{11}$$

where $\mathcal{U}$, $\mathcal{S}$, $\mathcal{V}$ derive from the tensor singular value decomposition (t-SVD) of $\phi_i(\mathcal{X}^k) - \frac{\Lambda_{v,i}^k}{\rho_i^v}$.

2) The subproblem of $\mathcal{X}$ at the k-th iteration is

$$\begin{aligned}\min_{\mathcal{X}}\ & \beta\|\mathcal{X}^k - \mathcal{W} - \frac{\Lambda_1^k}{\beta}\|_F^2 + \frac{\beta}{2}\|\mathcal{X} - \mathcal{Y}^k - \frac{\Lambda_2^k}{\beta}\|_F^2\\ &+ \frac{\beta}{2}\|\mathcal{A}(\mathcal{X}) - \mathcal{A}(\mathcal{Z}^k) - \frac{\Lambda_3^k}{\beta}\|_F^2 + \mu\Big(\frac{1}{2}\|\mathcal{A}(\mathcal{X}) - \mathcal{O}\|_F^2 + \frac{\rho}{2}\|\mathcal{X}\|_F^2\Big).\end{aligned} \tag{12}$$

Denoting $\mathcal{A}^*$ as the adjoint operator of $\mathcal{A}$, and letting $\mathcal{C} = \mu\mathcal{O} + \beta\mathcal{A}(\mathcal{Z}^k) + \Lambda_3^k$, $\mathcal{D} = \beta(\mathcal{W}^{k+1} + \mathcal{Y}^k) + \Lambda_1^k + \Lambda_2^k$, the solution of (12) can be calculated as

$$\mathcal{X}^{k+1} = \Big(\mathcal{Q}\beta + \mu\rho)\mathcal{I} + (\mu+\beta)\mathcal{A}^*\mathcal{A}\Big)^{-1}\big(\mathcal{A}^*(\mathcal{C}) + \mathcal{D}\big), \tag{13}$$

where $\mathcal{I}$ is the identity mapping operator and $(\cdot)^{-1}$ denote the inverse of an operator

3) The $\mathcal{Y}$ subproblem can be rewritten as

$$\min_{\mathcal{Y}} \frac{2}{\beta}\|\mathcal{Y} - \mathcal{X}^{k+1} + \frac{\Lambda_2^k}{\beta}\|_F^2 + \psi_1(\mathcal{Y}). \tag{14}$$

Under the PnP framework, the CNNs could serve as the proximal operator of regularization $\psi_1(\cdot)$, i.e., $\text{Prox}_{\psi_2} : \mathbb{R}^{n_1\times n_2\times n_3} \to \mathbb{R}^{n_1\times n_2\times n_3}$. Feeding $\mathcal{X}$ into the CNN pretrained for denoising, we obtain the solution of the $\mathcal{Y}$-subproblem as

$$\mathcal{Y}^{k+1} = \texttt{Denoising}_{\frac{1}{\beta}}\Big(\mathcal{X}^{k+1} - \frac{\Lambda_2^k}{\beta}\Big). \tag{15}$$

4) Similarly, the subproblem of $\mathcal{Z}$ can be formulated as

$$\min_{\mathcal{Z}} \psi_2(\mathcal{Z}) + \frac{\beta}{2}\|\mathcal{A}(\mathcal{Z}) - \mathcal{A}(\mathcal{X}^{k+1}) + \frac{\Lambda_3^k}{\beta}\|_F^2. \tag{16}$$

As discussed in [19], we can find that (16) is an inpainting problem and can be solved by plugging in an inpainting CNN as follows,

$$\mathcal{Z}^{k+1} = \texttt{Inpainting}\left(\mathcal{A}(\mathcal{X}^{k+1}) - \frac{\Lambda_3^k}{\beta}\right). \tag{17}$$

5) Finally, multipliers in our method are updated as

$$\begin{cases} \Lambda_1^{k+1} = \Lambda_1^k + \beta(\mathcal{W}^{k+1} - \mathcal{X}^{k+1}) \\ \Lambda_2^{k+1} = \Lambda_2^k + \beta(\mathcal{Y}^{k+1} - \mathcal{X}^{k+1}) \\ \Lambda_3^{k+1} = \Lambda_3^k + \beta(\mathcal{A}(\mathcal{Z}^{k+1}) - \mathcal{A}(\mathcal{X}^{k+1})). \end{cases} \tag{18}$$

The whole optimization scheme based on the ADMM framework is described in Algorithm 1.

Algorithm 1 ADMM for solving (7).

Input: $\mathcal{A}(\mathcal{O}), \lambda, \mu, \beta, \rho, k_{\max}, k = 0$

Initialization: : $\mathcal{W}^0 = \mathcal{Y}^0 = \mathcal{Z}^0 = \mathcal{X}^0 = \mathcal{A}(\mathcal{O})$, $\Lambda_1 = \Lambda_2 = \Lambda_3 = 0$.

1: **while** $k < k_{\max}$ **do**
2: Update $\mathcal{W}^{k+1}$ using (11).
3: Update $\mathcal{X}^{k+1}$ using (13).
4: Update $\mathcal{Y}^{k+1}$ using (15).
5: Update $\mathcal{Z}^{k+1}$ using (17).
6: Update $\{\Lambda_i\}_{i=1}^3$ using (18).
7: **end while**

Output: $\mathcal{W}$

5. Convergence analysis

Now we present the fixed-point analysis of Algorithm 1, whose outline follows the counterpart in [36]. We reformulate our optimization problem into (20), a similar formulation as presented in [36]. Building on this, we introduce the key assumptions for our proof, discuss the reasonableness, and present the main theorem in Section 5.2. Due to space constraints, the necessary lemmas and the detailed proof of the main theorem are provided in Supplementary Material.

5.1. Problem reformulation

Firstly, by the tensor vectorization operation, the model in (8) can be reformulated as

$$\min_{\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{w}} \lambda\|\mathbf{w}\|_\star + \psi_1(\mathbf{y}) + \psi_2(\mathbf{z}) + \mu\left(\frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{o}\|_F^2 + \frac{\rho}{2}\|\mathbf{x}\|_F^2\right), \quad \text{s.t.} \begin{bmatrix} \mathbf{w} \\ \mathbf{y} \\ \mathbf{A}\mathbf{z} \end{bmatrix} = \begin{bmatrix} \mathbf{x} \\ \mathbf{x} \\ \mathbf{A}\mathbf{x} \end{bmatrix}, \tag{19}$$

where $\mathbf{A}$ denotes the projection operator, $\mathbf{x}$,$\mathbf{y}$,$\mathbf{z}$, $\mathbf{w}$, and $\mathbf{o}$ are the vectorization of $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{Z}$, $\mathcal{W}$, and $\mathcal{O}$. Let

$$\mathbf{e} = \begin{bmatrix} \mathbf{w} \\ \mathbf{y} \\ \mathbf{z} \end{bmatrix}, \quad r(\mathbf{e}) = \begin{bmatrix} \lambda\|\mathbf{w}\|_\star \\ \psi_1(\mathbf{y}) \\ \psi_2(\mathbf{z}) \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} \mathbf{x} \\ \mathbf{x} \\ \mathbf{x} \end{bmatrix},$$

we can rewrite (19) as follows

$$\min_{\mathbf{e},\mathbf{v}} \quad \mu f(\mathbf{v}) + r(\mathbf{e}), \quad \text{s.t.} \quad \mathbf{e} = \mathbf{L}\mathbf{v}, \tag{20}$$

where $\mathbf{L} = \begin{bmatrix} \mathbf{I} & & \\ & \mathbf{I} & \\ & & \mathbf{A} \end{bmatrix}$, and $f(\cdot)$ denotes a scalar-valued function satisfying:

$$f(\mathbf{v}) = \frac{1}{2}\|\mathbf{A}[0\ 0\ \mathbf{I}]\mathbf{v} - \mathbf{o}\|_F^2 + \frac{\rho}{2}\|[0\ 0\ \mathbf{I}]\mathbf{v}\|_F^2 = \frac{1}{2}\|\mathcal{A}(\mathbf{x}) - \mathcal{O}\|_F^2 + \frac{\rho}{2}\|\mathbf{x}\|_F^2,$$

Then the augmented Lagrangian function of (20) can be written as

$$L(\mathbf{e},\mathbf{v},\boldsymbol{\lambda}) = \mu f(\mathbf{v}) + \frac{\beta}{2}\|\mathbf{L}\mathbf{e} - \mathbf{L}\mathbf{v} + \boldsymbol{\lambda}\|_2^2 + r(\mathbf{e}), \tag{21}$$

where $\boldsymbol{\lambda} = \begin{bmatrix} \frac{\lambda_1}{\beta} & \frac{\lambda_2}{\beta} & \frac{\lambda_3}{\beta} \end{bmatrix}^\top$ is the Lagrangian multiplier, $\{\boldsymbol{\lambda}_i\}_{i=1}^3$ are the vectorization of $\{\Lambda_i\}_{i=1}^3$ in (9), and without the loss of generality, the parameters relating to the multipliers are set to the same value denoted as β. Moreover, according to the standard steps of the ADMM, each variable is updated by iteratively solving corresponding subproblems shown as follow:

$$\begin{cases} \mathbf{e}^{k+1} = \arg\min_{\mathbf{e}} \left(\frac{\beta}{2}\left\|\mathbf{L}\mathbf{e} - \mathbf{L}\mathbf{v}^k + \boldsymbol{\lambda}^k\right\|_2^2 + r(\mathbf{e})\right) \\ \mathbf{v}^{k+1} = \arg\min_{\mathbf{v}} \left(\mu f(\mathbf{v}) + \frac{\beta}{2}\left\|\mathbf{L}\mathbf{e}^{k+1} - \mathbf{L}\mathbf{v} + \boldsymbol{\lambda}^k\right\|_2^2\right). \\ \boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \mathbf{L}\mathbf{e}^{k+1} - \mathbf{L}\mathbf{v}^{k+1} \end{cases} \tag{22}$$

Let denoting $\mathbf{P}_{\lambda\sigma} = \begin{bmatrix} \texttt{Prox}_\lambda & & \\ & \mathbf{D}_\sigma & \\ & & \mathbf{H} \end{bmatrix}$, where the $\texttt{Prox}_\lambda$ denotes the t-SVT operator in (11), $\mathbf{D}_\sigma(\cdot)$ is the CNN pretrained for denoising as in (15), $\mathbf{H}(\cdot)$ is the CNN for recovering utilized in (17). Then, based on the PnP framework, the corresponding solutions could be formulated as

$$\begin{cases} \mathbf{e}^{k+1} = \mathbf{P}_{\lambda\sigma}(\mathbf{L}\mathbf{v}^k - \boldsymbol{\lambda}^k) \\ \mathbf{v}^{k+1} = \texttt{Prox}_{\mu f}(\mathbf{L}\mathbf{e}^{k+1} + \boldsymbol{\lambda}^k), \\ \boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \mathbf{L}\mathbf{e}^{k+1} - \mathbf{L}\mathbf{v}^{k+1} \end{cases} \tag{23}$$

where $\texttt{Prox}_{\mu f}$ is the proximal operator of the subproblem of $\mathbf{v}$.

Similar to the interpretation in [36], we call the method formulated in (23) alternating directions method of multipliers for DAP+ (ADMM-DAP+) and say its fixed point is $(\mathbf{v}^\star, \{\boldsymbol{\lambda}_i^\star\}_{i=1}^3)$ if

$$\begin{cases} \mathbf{v}^\star = \mathbf{P}_{\lambda\sigma}(\mathbf{L}\mathbf{v}^\star - \boldsymbol{\lambda}^\star) \\ \mathbf{v}^\star = \texttt{Prox}_{\mu f}(\mathbf{v}^\star + \boldsymbol{\lambda}^\star) \end{cases}. \tag{24}$$

Let $\mathbf{e}^k = \mathbf{v}^k = \mathbf{v}^\star$ and $\boldsymbol{\lambda}_i^k = \boldsymbol{\lambda}_i^\star$ $(i = 1, 2, 3)$, then we can get $\mathbf{e}^{k+1} = \mathbf{v}^{k+1} = \mathbf{v}^\star$ and $\boldsymbol{\lambda}_i^{k+1} = \boldsymbol{\lambda}_i^k = \boldsymbol{\lambda}_i^\star$ $(i = 1, 2, 3)$. Furthermore, according to the equivalence of plug-and-play alternating directions method of multipliers (PNP-ADMM) and plug-and-play Douglas–Rachford splitting (PNP-DRS) verified in [36], let $\mathbf{u}^k = \mathbf{L}\mathbf{v}^k - \boldsymbol{\lambda}^k$, we can obtain the corresponding Douglas–Rachford splitting for DAP+ (DRS-DAP+) which is formulated as

$$\begin{cases} \mathbf{v}^{\frac{k+1}{2}} = \texttt{Prox}_{\mu f}(\mathbf{u}^k) \\ \mathbf{v}^{k+1} = \mathbf{P}_{\lambda\sigma}(2\mathbf{L}\mathbf{v}^{\frac{k+1}{2}} - \mathbf{u}^k) \\ \mathbf{u}^{k+1} = \mathbf{u}^k + \mathbf{L}\mathbf{v}^{k+1} - \mathbf{L}\mathbf{v}^{\frac{k+1}{2}} \end{cases} \tag{25}$$

In addition, we interpret this method as a fixed-point iteration and regard $\mathbf{u}^\star$ as its fixed point if

$$\begin{cases} \mathbf{v}^\star = \texttt{Prox}_{\mu f}(\mathbf{u}^\star) \\ \mathbf{v}^\star = \mathbf{P}_{\lambda\sigma}(2\mathbf{L}\mathbf{v}^\star - \mathbf{u}^\star) \end{cases}. \tag{26}$$

Equivalently, (25) can be rewriten as follows

$$\begin{cases} \mathbf{v}^{\frac{k+1}{2}} = \texttt{Prox}_{\mu f}(\mathbf{u}^k) \\ \mathbf{v}^{k+1} = \mathbf{P}_{\lambda\sigma}(2\mathbf{L}\mathbf{v}^{\frac{k+1}{2}} - \mathbf{u}^k) \\ \mathbf{u}^{k+1} = \mathbf{u}^k + \mathbf{L}\mathbf{P}_{\lambda\sigma}(2\mathbf{L}\mathbf{v}^{\frac{k+1}{2}} - \mathbf{u}^k) - \mathbf{L}\texttt{Prox}_{\mu f}(\mathbf{u}^k) \end{cases} \tag{27}$$

Now we can convert the iterations of $\mathbf{u}$ into:

$$\mathbf{u}^{k+1} = \frac{1}{2}\mathbf{u}^k + \frac{1}{2}(2\mathbf{L}\texttt{Prox}_{\mu f} - \mathbf{I})(2\mathbf{L}\mathbf{P}_{\lambda\sigma} - \mathbf{I})\mathbf{u}^k, \tag{28}$$

according to (25). As $\mathbf{L}$ is a operator, we denote that $\mathbf{B} = \mathbf{L}\mathbf{P}_{\lambda\sigma}$, Then (28) can be rewritten as

$$\mathbf{u}^{k+1} = \mathbf{M}(\mathbf{u}^k), \tag{29}$$

where $\mathbf{M} = \frac{1}{2}\mathbf{I} + \frac{1}{2}(2\mathbf{L}\texttt{Prox}_{\mu f} - \mathbf{I})(2\mathbf{B} - \mathbf{I})$, and the subsequent analysis will be carried out by studying the contractiveness of the operator $\mathbf{M}$.

5.2. Key assumption and main theorem

It should be noted that if only a denoiser is considered in our method, the convergence analysis of Algorithm 1 would extremely follow the corresponding part in [36], but the primary difference between this work and typical PnP-based methods makes the analysis more complex, so new assumptions is required, and we need to verify their reasonability.

We **assume B** $= \begin{bmatrix} \texttt{Prox}_\lambda & & \\ & \mathbf{D}_\sigma & \\ & & \mathbf{LH} \end{bmatrix} : \mathbb{R}^d \to \mathbb{R}^d$ satisfies

$$\|(\mathbf{B}-\mathbf{I})(\mathbf{x}_1)-(\mathbf{B}-\mathbf{I})(\mathbf{x}_2)\|^2 \le \epsilon^2\|\mathbf{x}_1-\mathbf{x}_2\|_2^2, \tag{A}$$

for all $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^d$ for some $\epsilon \ge 0$. Subsequently, it is rational to say that **LH** is close to the identity when the recovering CNNs are well-trained. Thirdly, $\texttt{Prox}_\lambda$ is essentially a t-SVT operator (although it is directly applied to the third-order tensor, it can also be viewed as a mapping: $\mathbb{R}^d \to \mathbb{R}^d$ with the implicit employment of reshape transformations, whose operator norms are equal to 1, between the vector and the tensor. Here the vectorization and its inverse operation are denoted as vec$(\cdot)$ and vec$^{-1}(\cdot)$, respectively), which can be formulated as:

$$\texttt{Prox}_\lambda(\mathcal{X}) = \text{t-SVT}_\lambda(\mathcal{X}) \triangleq \mathcal{U} * \mathcal{D} * \mathcal{V}^H, \tag{30}$$

where $\mathcal{U}, \mathcal{S}, \mathcal{V}^H$ are attained by the t-SVD of $\mathcal{X}$, and $\mathcal{D}$ is an f-diagonal tensor obtained satisfying $\mathcal{D}(i,i,k) = \max\{\tilde{\mathcal{S}}(i,i,k)-\lambda, 0\}$, where $\tilde{\mathcal{S}}$ is the Discrete cosine transformed (along the third mode) tensor of $\mathcal{S}$. By the Lemma 3.19 in [9], we can write that

$$\|\mathcal{X}\|_F^2 = \|\mathcal{U} * \mathcal{S} * \mathcal{V}^H\|_F^2 = \|\mathcal{S}\|_F^2 = \left\| \begin{bmatrix} \mathcal{S}_{(1)} & & & \\ & \mathcal{S}_{(2)} & & \\ & & \cdots & \\ & & & \mathcal{S}_{(n_3)} \end{bmatrix} \right\|_F^2, \tag{31}$$

where $\mathcal{S}_{(i)}$'s $(i = 1, 2, 3, \ldots, n_3)$ are the i-th frontal slices of $\mathcal{S}$, $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$. Similarly, by the unitary invariance of Frobenius norm, we denotes the $\mathbf{C}_n$ as DCT matrix and write as :

$$\begin{aligned} \|\mathcal{X}\|_F^2 &= \left\| \mathbf{C}_{n_3} \otimes I \begin{bmatrix} \mathcal{X}_{(1)} & & & \\ & \mathcal{X}_{(2)} & & \\ & & \cdots & \\ & & & \mathcal{X}_{(n_3)} \end{bmatrix} \mathbf{C}_{n_3}^* \otimes I \right\|_F^2 \\ &= \left\| \begin{bmatrix} \tilde{\mathcal{X}}_{(1)} & & & \\ & \tilde{\mathcal{X}}_{(2)} & & \\ & & \cdots & \\ & & & \tilde{\mathcal{X}}_{(n_3)} \end{bmatrix} \right\|_F^2 = \left\| \begin{bmatrix} \tilde{\mathcal{S}}_{(1)} & & & \\ & \tilde{\mathcal{S}}_{(2)} & & \\ & & \cdots & \\ & & & \tilde{\mathcal{S}}_{(n_3)} \end{bmatrix} \right\|_F^2 \end{aligned} \tag{32}$$

where $\tilde{\mathcal{X}}$ is the discrete cosine transformed (along the third mode) tensor of $\mathcal{X}$. Then we sort all the singular values in descending order of all $\tilde{\mathcal{X}}_{(i)}$'s as $s_1 \ge s_2 \ge \cdots \ge s_{k_1} \ge \lambda \ge \cdots \ge s_{k_2} > s_{k_2+1} = 0$ and consider the unitary invariance of orthogonal matrix, then we can directly have

$$\left\| \begin{bmatrix} \tilde{\mathcal{S}}_{(1)} & & & \\ & \tilde{\mathcal{S}}_{(2)} & & \\ & & \cdots & \\ & & & \tilde{\mathcal{S}}_{(n_3)} \end{bmatrix} \right\|_F^2 = \left\| \begin{bmatrix} s_1 & & & & \\ & s_2 & & & \\ & & s_3 & \cdots & \\ & & & & s_{k_2} \end{bmatrix} \right\|_F^2 \tag{33}$$

and

$$\|\texttt{Prox}_\lambda(\mathcal{X})\|_F^2 = \left\| \begin{bmatrix} s_1-\lambda & & & & & \\ & \cdots & & & & \\ & & s_{k_1}-\lambda & & & \\ & & & 0 & \cdots & \\ & & & & & 0 \end{bmatrix} \right\|_F^2. \tag{34}$$

Then it is easy to obtain

$$\|(\texttt{Prox}_\lambda - \mathbf{I})(\mathcal{X})\|_F^2 = \left\| \begin{bmatrix} -\lambda & & & & & \\ & \cdots & & & & \\ & & -\lambda & & & \\ & & & -s_{k_1+1} & & \\ & & & & \cdots & \\ & & & & & -s_{k_2} \end{bmatrix} \right\|_F^2. \tag{35}$$

It is evident that $\texttt{Prox}_\lambda$ will be close to identity with λ close to 0. Furthermore, by reshaping $\mathcal{X}_1$ and $\mathcal{X}_2$ into the vector $\mathbf{x}_1$ and $\mathbf{x}_2$ and utilizing the ℓ_2 norm of the vectors, (35) can be formulated as:

$$\|(\texttt{Prox}_\lambda - \mathbf{I})(\mathbf{x}_1)\|_2 = \sqrt{k_1\lambda^2 + \sum_{k=k_1+1}^{k_2} (s_k^x)^2}, \tag{36}$$

and for $\mathbf{x}_2$:

$$\|(\texttt{Prox}_\lambda - \mathbf{I})(\mathbf{x}_2)\|_2 = \sqrt{l_1\lambda^2 + \sum_{l=l_1+1}^{l_2} (s_l^y)^2}. \tag{37}$$

where $(k_1, k_2, \{s_k^x\})$, $(l_1, l_2, \{s_l^y\})$ are respectively the numbers of singular values of $\mathbf{x}_1$ and $\mathbf{x}_2$ that are greater than λ_i and 0, as well as the k/l-th $(k/l = 1, \cdots, k_2/l_2)$ largest singular values, and we can directly get:

$$\|\mathbf{x}_1\|_2 = \sqrt{\sum_{k=1}^{k_2} (s_k^x)^2}, \quad \|\mathbf{x}_2\|_2 = \sqrt{\sum_{l=1}^{l_2} (s_l^y)^2}. \tag{38}$$

According to the triangle inequality:

$$\left(\|\mathbf{x}_1\|_2 - \|\mathbf{x}_2\|_2\right)^2 \le \|\mathbf{x}_1 - \mathbf{x}_2\|_2^2 \le \left(\|\mathbf{x}_1\|_2 + \|\mathbf{x}_2\|_2\right)^2, \tag{39}$$

then

$$\|(\texttt{Prox}_\lambda - \mathbf{I})(\mathbf{x}_1) - (\texttt{Prox}_\lambda - \mathbf{I})(\mathbf{x}_2)\|_2^2 \le \left(\|(\texttt{Prox}_\lambda - \mathbf{I})(\mathbf{x}_1)\|_2 + \|(\texttt{Prox}_\lambda - \mathbf{I})(\mathbf{x}_2)\|_2\right)^2.$$

Therefore, the third part of Assumption (A):

$$\|(\texttt{Prox}_\lambda - \mathbf{I})(\mathbf{x}_1) - (\texttt{Prox}_\lambda - \mathbf{I})(\mathbf{x}_2)\|_2^2 \le \epsilon^2\|\mathbf{x}_1 - \mathbf{x}_2\|_2^2 \tag{40}$$

will be hold if the λ, k_1, l_1, $\{s_k^x\}(k = 1, 2, \cdots, k_2)$ and $\{s_l^y\}(l = 1, 2, \cdots, l_2)$ satisfy the following inequality:

$$\begin{aligned} &\left(\sqrt{k_1\lambda^2 + \sum_{k=k_1+1}^{k_2} (s_k^x)^2} + \sqrt{l_1\lambda^2 + \sum_{l=l_1+1}^{l_2} (s_l^y)^2}\right)^2 \le \epsilon^2 \left(\sqrt{\sum_{k=1}^{k_2} (s_k^x)^2} - \sqrt{\sum_{l=1}^{l_2} (s_l^y)^2}\right)^2. \\ &\Leftrightarrow \left(\|(\texttt{Prox}_\lambda - \mathbf{I})(\mathbf{x}_1)\|_2 + \|(\texttt{Prox}_\lambda - \mathbf{I})(\mathbf{x}_2)\|_2\right)^2 \le \epsilon^2 (\|\mathbf{x}_1\|_2 - \|\mathbf{x}_2\|_2)^2 \end{aligned} \tag{41}$$

The Assumption (A) is reasonable based on the above discussion.

Theorem 2 (Convergence of DAP+). *Assume that* **B** *satisfies assumption* (A) *for some* $\epsilon \ge 0$. f *in* (20) *is* ρ*-strongly convex and differentiable. Then*

$$\mathbf{M} = \frac{1}{2}\mathbf{I} + \frac{1}{2}(2\mathbf{L}Prox_{\mu f} - \mathbf{I})(2\mathbf{B} - \mathbf{I}) \tag{42}$$

satisfies

$$\|\mathbf{M}(\mathbf{x}_1) - \mathbf{M}(\mathbf{x}_2)\| \le \frac{1 + \epsilon + \epsilon\mu\rho + 2\epsilon^2\mu\rho}{1 + \mu\rho + 2\epsilon\mu\rho}\|\mathbf{x}_1 - \mathbf{x}_2\|$$

for all $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^d$. *The coefficient is less than* 1 *if*

$$\frac{\epsilon}{(1 + \epsilon - 2\epsilon^2)\rho} < \mu, \quad \epsilon < 1.$$

Owing to the limitation of space, the proof of Theorem 2 is presented in Supplementary Material.

6. Experiments results

With a series of experiments, we assess the performance of the proposed degradation accordant plug-and-play for LRTR (DAP+) and compare it with other state-of-the-art methods. **Compared methods**[1] are: a Tucker decomposition based-method, HaLRTC [23]; a t-SVD-based method that utilizes the tensor nuclear norm as the convex surrogate for the tensor tubal rank, TNN [10]; a dictionary-learning-induced method

[1] As some compared methods only consider noise-free tensor completion. We remark here that it is necessary and easy to modify their methods for the noisy tensor completion problem by replacing the indicator function in their objective function with the weighted Frobenius norm. Meanwhile, the hard projection step of the intermediate variable should be correspondingly changed to be in the same manner as (13).

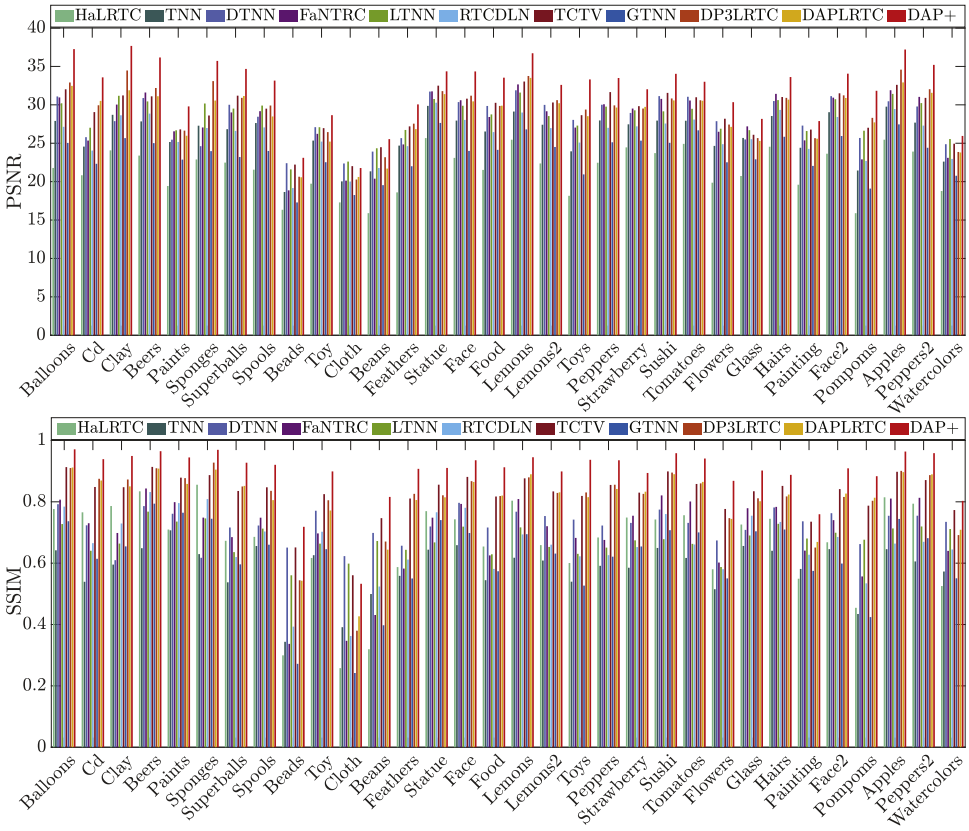

Fig. 3. The PSNR and SSIM values of results by various methods on CAVE dataset (SR = 0.05 & σ = 10.)

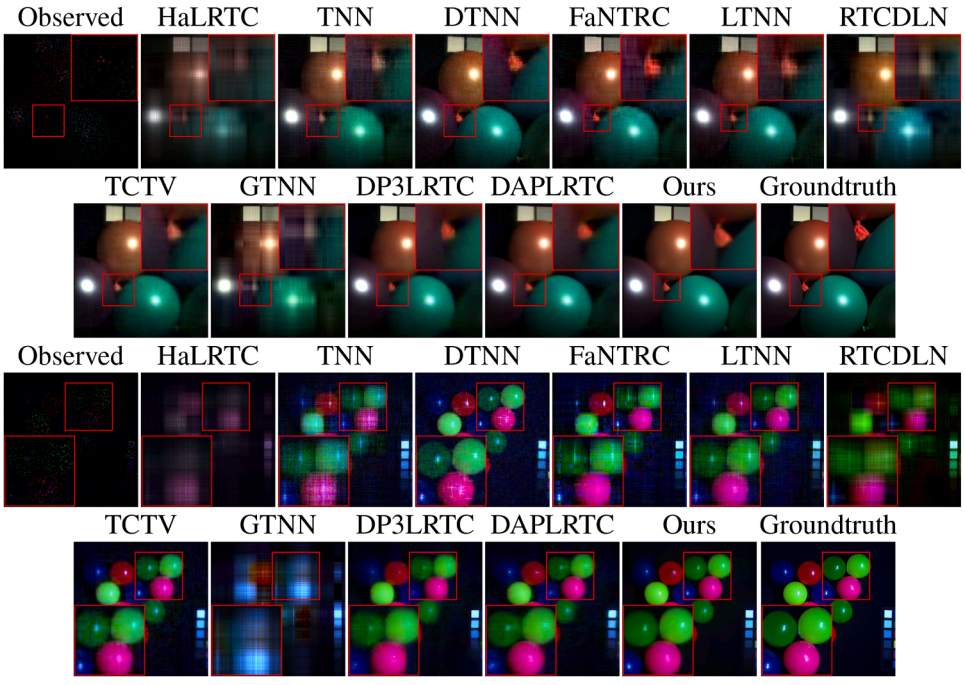

Fig. 4. Pseudo color images of restored results by various methods on MSIs *Balloons* and *Superballs* (SR = 5 % & σ = 10).

based on the t-SVD framework, DTNN[25]; a tensor ring decomposition-based method with high computational efficiency, FaNRTC [20]; a t-SVD-based method with a balanced-unfolding strategy, LTNN [13]; a method combining nonconvex regularization and dictionary learning within the t-SVD framework, RTCDLN [12]; a t-SVD-based method that generates sparsity-inducing regularizers with closed-form thresholding functions, GTNN [37]; a method that exploits local smoothness as well as the lowr-rankness based the high-order t-SVD framework, TCTV [4]; a PnP-based method that combines the tensor low-rank prior and a data-driven prior by utilizing a deep denoiser, DP3LRTC [17]; and a PnP-based method that combines the tensor low-rank prior and a data-driven prior by utilizing a deep inpainter, DAPLRTC [19]. As some of the

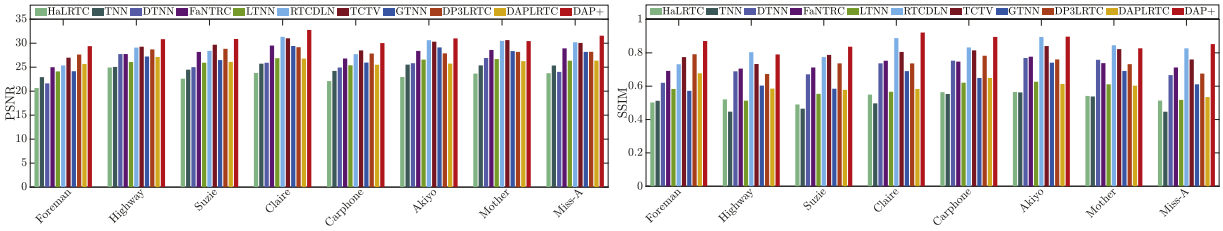

Fig. 5. The PSNR and SSIM values of results by various methods on 8 color videos (SR = 0.2 & σ = 20).

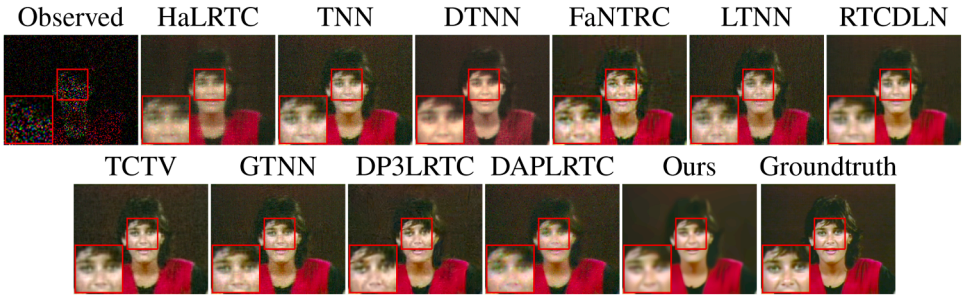

Fig. 6. Visual results by various methods on the color video *Miss A* (SR = 20 % & σ = 20).

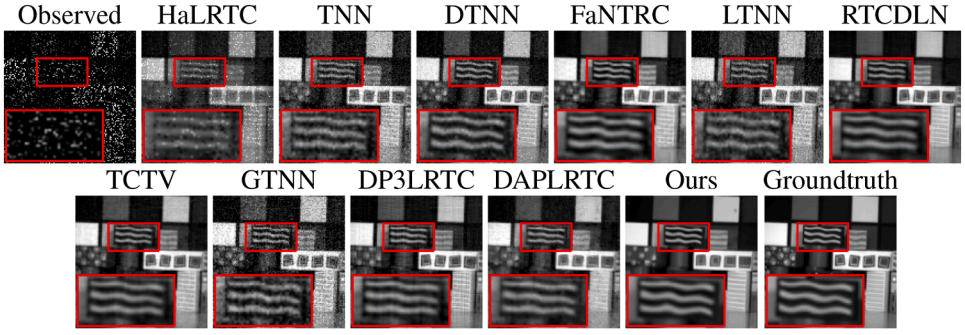

Fig. 7. The restored results by various methods on the HSV data (SR = 20 % & σ = 20).

compared methods are originally designed for the noise-free case, we have modified the constraint in their model with the Gaussian noise taken into account for fairness.

The multidimensional visual datasets, including multispectral images (MSIs) from the CAVE dataset[2], color images from the Urban100 dataset [14], color videos from the Yuv dataset[3], and hyper-spectral video (HSV)[4] are chosen to evaluate the performance of different methods. Additionally, the Supplementary Material provides experimental results on the grayscale videos from the Yuv dataset and demonstrates the practical value of our method with applications in compressive sensing magnetic resonance imaging (CS-MRI).

We remark here that different types of data for experiments in this work are not included in the training set of the denoiser or inpainter employed in our algorithm. It should also be noted that we reshape the color videos (CV) and HSV data into third-order tensors in the corresponding experiments.

For MSIs data, we consider three numerical metrics: the Peak signal-to-noise ratio (PSNR), the structural similarity index (SSIM), and the mean spectral angle mapper (SAM), to evaluate the reconstructed result of all the methods. For other data, PSNR and SSIM are utilized to measure the effectiveness of each method, and except for the color image and MSI, the gradient magnitude similarity deviation (GMSD) [38] is also utilized. Higher PSNR, SSIM values and lower SAM, GMSD values indicate better performance. As the PSNR, SSIM, and GMSD are designed to measure the quality of natural images, we compute them slice by slice for our multidimensional visual data and then report the average value.

A CNN denoiser is considered in our method: DRUNet in [18], and two CNN inpainters are employed in our method: i) CRUNet in [19] for random missing, ii) LBAM [39] for structural missing. All experiments were conducted on the platform of Windows 10 with an RTX 2080Ti GPU, AMD Ryzen9 3950X CPU, and 32RAM.

6.1. Parameter setting

Referring to the augmented Lagrangian function of our model in (9), the parameters that need to be specified are λ, μ, β and ρ, then let the thresholding $\gamma = \lambda/\beta$, where γ can be singly used for solving the subproblem of $\mathcal{W}$. In all experiments, ρ is set to 10^{-4}. For MSIs, grayscale videos and color videos, (λ, μ, β) are set to (1,1,1), and $\gamma \in [0.5, 0.7]$,$[0.1, 0.4]$, and $[0.9, 1]$ respectively. For HSV data, (λ, μ, β) are set to (1,10,10) and $\gamma \in [5, 10]$. Meanwhile we set (λ, μ, β) to (1,1,1) and $\gamma = 0.1$.

6.2. Multi-spectral image (MSI)

The CAVE dataset with 32 MSIs of the size $256 \times 256 \times 31$ is considered in this part. The observation tensors are obtained by i) adding zero-mean Gaussian noise with different standard deviations (σ) and ii) uniformly random sampling with different sampling rates (SRs), respectively. As MSIs are highly inner correlated, we test all the methods

[2] http://www.cs.columbia.edu/CAVE/databases/multispectral/
[3] http://trace.eas.asu.edu/yuv/
[4] http://openremotesensing.net/knowledgebase/hyperspectral-video/

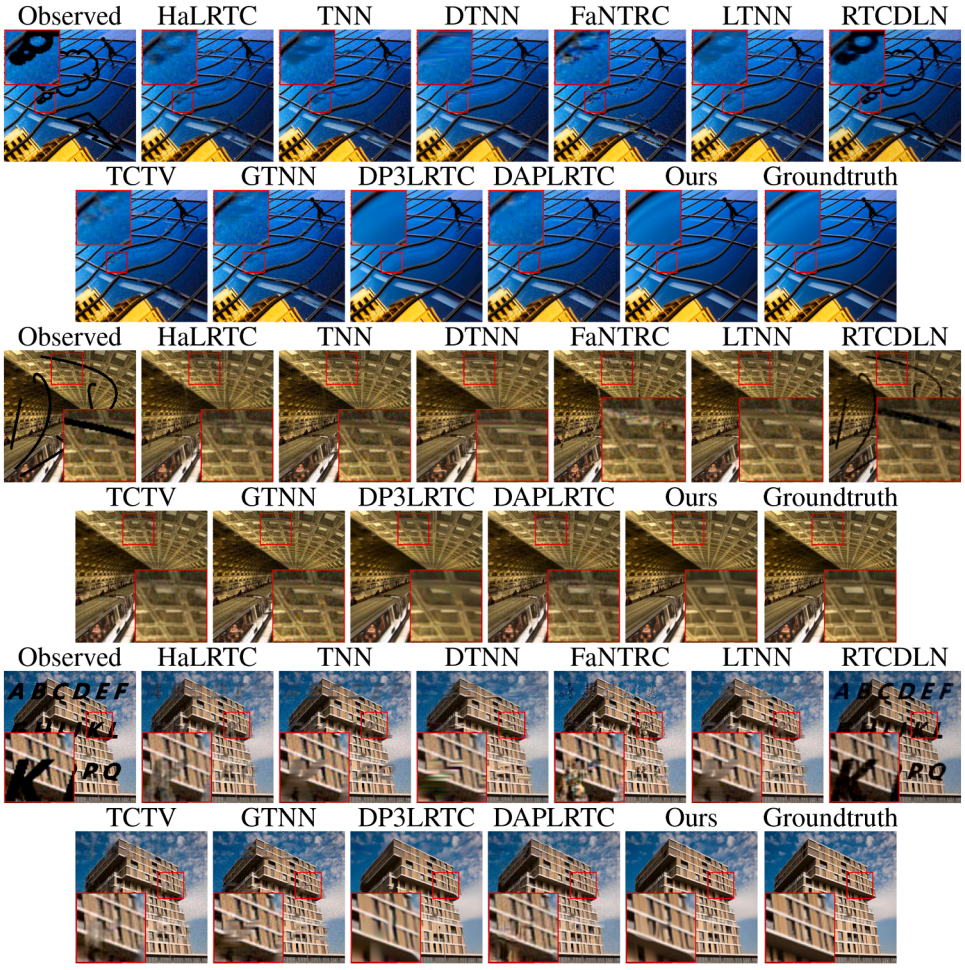

Fig. 8. The recovered results by different methods on the color images (#82, #84, and #87 in Urban100 datasets) with structural missing and noisy level $\sigma = 15$.

Table 2
Quantitative results of various methods averaged across 32 multispectral images (MSIs) with varying SRs and σs. The **best** and second best values are highlighted by boldface and underline, respectively.

SR & σ	3% & 5			5% & 10			10% & 15			Time
Method	PSNR↑	SSIM↑	SAM↓	PSNR↑	SSIM↑	SAM↓	PSNR↑	SSIM↑	SAM↓	
Observed	14.39	0.197	-	14.47	0.197	-	14.65	0.183	-	-
HaLRTC	19.20	0.600	29.898	21.52	0.662	26.148	23.80	0.705	23.693	28.3
TNN	23.91	0.590	30.958	26.12	0.585	31.494	26.22	0.494	32.767	18.39
DTNN	27.03	0.726	26.974	28.18	0.724	26.386	29.73	0.738	25.155	252.0
FaNTRC	26.21	0.663	23.998	27.87	0.696	20.565	28.41	0.711	20.728	**11.3**
LTNN	27.89	0.745	18.064	28.35	0.677	22.625	27.42	0.773	17.037	113.6
RTCDLN	24.23	0.608	30.628	26.12	0.663	27.269	28.42	0.740	21.866	279.9
TCTV	27.98	0.808	20.810	29.09	0.827	20.580	30.30	0.835	19.771	128.5
GTNN	21.43	0.574	35.734	23.53	0.617	33.660	26.23	0.672	30.468	48.3
DP3LRTC	28.28	0.822	10.521	29.23	0.809	13.999	29.24	0.731	20.292	186.1
DAPLRTC	27.39	0.812	11.418	28.66	0.806	13.668	29.09	0.731	19.067	25.4
DAP+	**29.73**	**0.872**	**9.595**	**32.14**	**0.897**	**8.159**	**33.79**	**0.915**	**6.848**	90.2

with some challenging combinations of the SR and σ and we list average quantitative metrics of the results by different methods in Table 2. We also report PSNR and SSIM values of the results by different methods on 32 MSIs with SR $= 0.05$, $\sigma = 10$ in Fig. 3. Then we concatenate the 25-th, 15-th, and 8-th bands of results by all methods on the MSIs (*Balloons*, *Superballs*) to get Pseudo-color images and show them in Fig. 4. It can be observed that the proposed method demonstrates superior performance among these methods, owing to its inherent strength to simultaneously reconstruct both high-frequency and low-frequency information. Consequently, the missed structural information is accurately reconstructed, and the noise artifacts are effectively mitigated, as demonstrated in Fig. 4. Conversely, it is evident that methods primarily leveraging tensor low-rank priors exhibit limitations in noise removal and the accurate recovery of geometric information. DAPLRTC struggles with the noise corruption, whereas DP3LRTC generates overly-smooth results, which can be attributed to their incomplete consideration of the degradation within the PnP framework.

6.3. Color video

8 color videos of the size $144 \times 176 \times 3$ (color) $\times 50$ (frame) are selected to measure the performance of different methods. Firstly, we

Table 3
Quantitative results of various methods averaged across 8 color videos with varying SRs and σs. The **best** and second best values are highlighted by bold and underlined, respectively.

SR & σ	5% & 10			10% & 15			20% & 20			Time (s)
Method	PSNR↑	SSIM↑	GMSD↓	PSNR↑	SSIM↑	GMSD↓	PSNR↑	SSIM↑	GMSD↓	
Observed	6.36	0.019	-	6.59	0.026	-	7.07	0.037	-	-
HaLRTC	19.43	0.574	0.156	21.67	0.592	0.126	23.06	0.531	0.105	51.7
TNN	20.57	0.508	0.123	25.96	0.615	0.080	24.84	0.503	0.089	101.0
DTNN	21.89	0.702	0.184	22.86	0.715	0.146	25.25	0.708	0.091	367.6
FaNTRC	26.16	0.692	0.100	26.81	0.684	0.092	27.90	0.730	0.083	**41.3**
LTNN	23.73	0.609	0.116	25.94	0.609	0.087	26.01	0.574	0.086	768.4
RTCDLN	24.38	0.723	0.132	27.01	0.786	0.095	29.15	0.825	0.069	485.4
TCTV	25.04	0.772	0.100	27.97	0.782	0.077	29.56	0.792	0.071	275.8
GTNN	25.29	0.713	0.113	27.53	0.764	0.085	27.37	0.643	0.086	61.2
DP3LRTC	25.10	0.738	0.106	28.24	0.822	0.071	28.31	0.736	0.074	194.9
DAPLRTC	25.85	0.761	0.085	26.89	0.723	0.074	26.20	0.603	0.077	101.9
DAP+	**28.61**	**0.838**	**0.068**	**30.14**	**0.861**	**0.061**	**30.82**	**0.862**	**0.056**	65.4

Table 4
Quantitative results of all methods on the HSV data with varying SRs and σs. The **best** and second best values are highlighted via boldface and underline, respectively.

SR & σ	5% & 10			10% & 15			20% & 20			Time
Method	PSNR↑	SSIM↑	GMSD↓	PSNR↑	SSIM↑	GMSD↓	PSNR↑	SSIM↑	GMSD↓	
Observed	9.18	0.020	-	9.40	0.034	-	9.87	0.058	-	-
HaLRTC	15.24	0.428	0.228	19.10	0.564	0.154	21.62	0.583	0.110	**69.7**
TNN	22.39	0.677	0.121	26.85	0.738	0.081	25.31	0.642	0.084	151.7
DTNN	26.31	0.834	0.086	27.52	0.819	0.072	27.30	0.752	0.070	1338.5
FaNTRC	28.51	0.860	0.074	29.36	0.878	0.064	29.59	0.887	0.060	38.0
LTNN	23.53	0.712	0.128	25.89	0.702	0.097	25.41	0.653	0.093	767.1
RTCDLN	25.01	0.807	0.146	27.97	0.875	0.103	30.94	0.913	0.073	774.5
TCTV	28.19	0.871	0.079	30.23	0.889	0.063	31.56	0.904	0.054	353.3
GTNN	27.42	0.833	0.105	29.93	0.852	0.072	24.80	0.613	0.092	89.8
DP3LRTC	28.21	0.868	0.087	30.24	0.875	0.070	30.66	0.872	0.064	387.8
DAPLRTC	28.76	0.875	0.074	29.89	0.865	0.071	29.73	0.856	0.063	226.5
DAP+	**30.11**	**0.907**	**0.065**	**31.54**	**0.926**	**0.053**	**32.74**	**0.938**	**0.046**	130.1

reshape these data into tensors of the size $144 \times 176 \times 150$. It can be observed from Table 3 that our method obtains the best metrics for all cases. Fig. 5 displays the PSNR and SSIM values of the results by different methods on color video data with SR $= 0.2$, $\sigma = 20$. Fig. 6 shows the reconstructed results by different methods on the color video *Miss-A* with SR $= 0.2$ and $\sigma = 20$, and it is evident that our method obtains a clear result. Results by other methods, even including the deep denoiser-based method DP3, are not satisfactory for remaining noise or inferior completion effectiveness.

6.4. Hyper-spectral video (HSV)

In this experiment, a hyper-spetral video[5] of size $480 \times 752 \times 20(\text{band}) \times 20$ (frame) is selected as a 4D data to test the strength of different methods. Considering the computational limitation, all spatial slices of HSV are downsampled by us, and then we reshape it into a smaller tensor of size $120 \times 120 \times 400$. The sampling rates (SR) and noise level σ are the same as in the experiments on color videos and the quantitative results and visual results by all considered methods are listed in Table 4 and Fig. 7, respectively. It is explicit that our method outperforms other methods both quantitatively and visually.

6.5. Color image with structural missing

In this part, we test all the methods on color images from Urban100 datasets [14] with structural missing and Gaussian noise corruptions with the noise level $\sigma = 15$ in all RGB channels. Table 5 exhibits the quantitative metrics by different methods on color images (#68,#82, #84, #87 in Urban100 datasets) with different types of missing areas. Fig. 8 shows the corresponding reconstructed results on the color images (#82, #84, #87) by all methods.

It could be obviously observed that our method made more accurate results in more intricate structural-missing situations, and DP3LRTC fails to the structural information while DAPLRTC struggles hard with noises.

6.6. Ablation study

Our method consists of three important modules, including the low-rankness (LR) part, denoiser (Dn), and inpainter (Ipt). We conduct the ablation study by testing different combinations of these three regularizers on whole MSIs in CAVE dataset with SR $= 5\%$ and noise level $\sigma = 10$. We can see that our method without the low-rankness regularization, which considers the decoupled data degradation process, could also generate a good result, and obviously, results of the model when all priors are considered outperform other cases, which could clearly demonstrate the effect of our model. Meanwhile, we also vary the low-rank regularization with LRMF[40], HaLRTC[23], TNN[10], DCT-based TNN (DCTNN) [11] to examine the effectiveness of different low-rank tensor regularizers. We show all quantitative metrics in Table 6, the results by the model incorporating the considered DCTNN in this work have an improvement over others.

6.7. Parameter analysis and convergence behaviour

We conduct experiments on the grayscale video named *Carphone* with SR $= 10\%$ and $\sigma = 15$ to test effects from different values of parameters β and μ, which affect the performance at most. In Fig. 9, We report the PSNR and SSIM values of the restored results for different β/μ, with all other parameters held constant. Obviously, the effectiveness of our method is sensitive to β and μ, and it would be better to adjust their

[5] http://openremotesensing.net/knowledgebase/hyperspectral-video/

Table 5
Quantitative results by various methods on color images (#68,#82, #84, and #87 in Urban100 datasets) with different structural missing areas and noise level $\sigma = 15$. The **best** and second best values are highlighted by boldface and underline, respectively.

Mask Type	Type1		Type2		Type 3		Type 4		Time (s)
Method	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	
Observed	12.81	0.465	17.03	0.567	18.44	0.619	14.39	0.552	-
HaLRTC	12.88	0.488	23.44	0.666	24.23	0.730	21.40	0.669	6.2
TNN	21.38	0.619	22.97	0.625	23.09	0.680	21.03	0.624	**0.4**
DTNN	22.46	0.670	25.36	0.729	24.66	0.771	22.07	0.710	157.5
FaNTRC	12.88	0.493	22.98	0.686	23.39	0.746	20.32	0.682	4.6
LTNN	22.07	0.653	24.12	0.665	24.03	0.725	21.78	0.663	7.4
RTCDLN	12.80	0.516	19.07	0.697	20.47	0.752	15.04	0.661	89.3
TCTV	23.43	0.729	25.48	0.772	27.30	0.837	23.81	0.788	10.0
GTNN	22.16	0.641	23.85	0.638	23.41	0.692	22.93	0.670	3.7
DP3LRTC	21.68	0.786	29.39	0.952	28.34	0.869	24.17	0.883	47.1
DAPLRTC	23.97	0.728	25.38	0.741	26.43	0.796	23.89	0.762	3.0
DAP+	**27.29**	**0.908**	**32.37**	**0.970**	**30.66**	**0.933**	**26.87**	**0.928**	18.1

Table 6
Quantitative results by various combinations of different terms across 32 MSIs in the Cave dataset with SR = 5 % and $\sigma = 10$. The **best** values are highlighted in bold.

LR	DCTNN [11]			DCTNN	DCTNN		LRMF [40]	HaLRTC [23]	TNN [10]	DCTNN (Ours)
DN		√		√		√	√	√	√	√
Ipt			√		√	√	√	√	√	√
PSNR↑	27.33	28.03	29.91	29.20	30.63	31.23	31.45	31.95	31.91	**32.14**
SSIM↑	0.739	0.759	0.746	0.802	0.793	0.867	0.871	0.888	0.889	**0.897**
SAM ↓	23.313	12.446	18.067	9.853	16.478	10.010	10.410	8.721	8.992	**8.159**

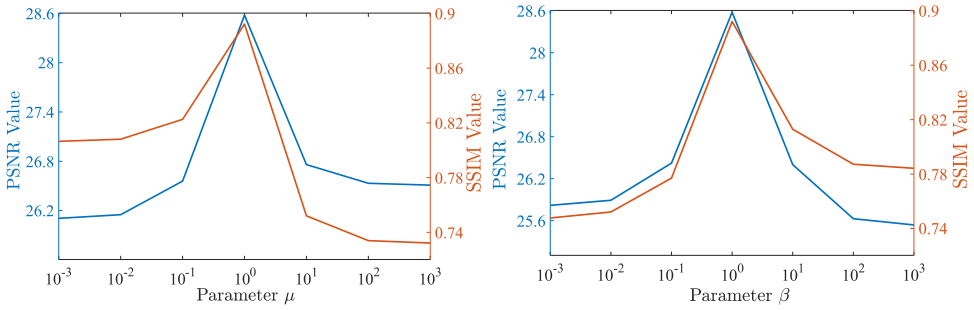

Fig. 9. The PSNR and SSIM values of results by our method with different μ, β on the Video *Carphone* (SR = 10 %, $\sigma = 15$).

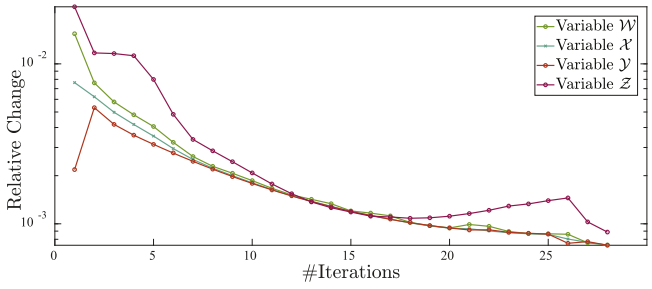

Fig. 10. The relative changes of the variables $\mathcal{W}$, $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{Z}$ on the video data *Carphone* with SR = 5 % and $\sigma = 10$.

values near 1. Meanwhile, in Fig. 10, we plot relative changes of variables in our algorithm on the video data *Carphone* with SR = 5 % and $\sigma = 10$. We could not only theoretically establish the convergence of our model, but also empirically see from Fig. 10 that all variables converge rapidly and meet the stop criteria at the 28-th iteration (the convergence criteria is set to 10^{-3}).

7. Conclusions

We propose a novel low-rank noisy tensor completion method for multi-dimensional visual data. First, the DCT-based TNN is adopted to characterize the global high-correlated structure of tensors. Then, we resort to the plug-and-play framework to introduce data-driven priors expressed by convolutional neural networks. Unlike previous methods using CNN denoiser, we decouple the degradation into two parts, i.e., the loss of data values and the impact of noise, by formulating two implicit regularizers in our model. Thus, after developing an ADMM-based algorithm, it is convenient for us to plug in a CNN denoiser and a CNN inpainter with degradation accordance. Furthermore, the convergence of the solution algorithm is theoretically substantiated. Numerical experiments are conducted on various types of multi-dimensional images to demonstrate that our method can outperform state-of-the-art methods, additionally, the great performance on CS-MRI demonstrates the proposed model's ability to generalize across other applications for multi-dimensional imaging data. However, for tasks involving different degradation processes, the proposed approach still requires reformulating the model and re-designing the solving algorithm. Additionally, compared to existing methods with supervised learning based deep

neural networks, the runtime of the proposed method is slower, which constitutes a key limitation in applications requiring real-time processing. In the future, we will focus on generalizing this framework to other tensor recovery tasks, not limited to imaging data. Furthermore, improving the computational efficiency of our method is also an important consideration for future research.

CRediT authorship contribution statement

Yexun Hu: Writing – original draft, Validation, Methodology; **Zixin Tang:** Writing – review & editing, Validation, Investigation, Conceptualization; **Tai-Xiang Jiang:** Writing – review & editing, Supervision, Project administration, Funding acquisition; **Xi-Le Zhao:** Writing – review & editing, Resources; **Guisong Liu:** Writing – review & editing.

Data availability

All of the data used are public available. We have provide the associated website links when using them.

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

This work was supported in part by the Sichuan Science and Technology Program under Grant 2024ZYD0147, Grant 2024NSFSC1452, Grant 2024NSFJQ0038, and Grant 2024NSFSC0038; in part by the Natural Science Foundation of Xinjiang Uygur Autonomous Region under Grant 2024D01A18; in part by the Guanghua Talent Project; in part by the Doctoral Research Project of the Research Institute for Digital Economy and Interdisciplinary Sciences of Southwestern University of Finance and Economics.

References

[1] Y. Chang, Z. Tu, W. Xie, B. Luo, S. Zhang, H. Sui, J. Yuan, Video anomaly detection with spatio-temporal dissociation, Pattern Recognit. 122 (2022) 108213.
[2] W. He, Y. Chen, N. Yokoya, C. Li, Q. Zhao, Hyperspectral super-resolution via coupled tensor ring factorization, Pattern Recognit. 122 (2022) 108280.
[3] N. Weiskopf, L.J. Edwards, G. Helms, S. Mohammadi, E. Kirilina, Quantitative magnetic resonance imaging of brain anatomy and in vivo histology, Nat. Rev. Phys. 3 (8) (2021) 570–588.
[4] H. Wang, J. Peng, W. Qin, J. Wang, D. Meng, Guaranteed tensor recovery fused low-rankness and smoothness, IEEE Trans. Pattern Anal. Mach. Intell. 45 (9) (2023) 10990–11007.
[5] S. Yuan, K. Huang, A generalizable framework for low-rank tensor completion with numerical priors, Pattern Recognit. 155 (2024) 110678.
[6] L.R. Tucker, Some mathematical notes on three-mode factor analysis, Psychometrika 31 (3) (1966) 279–311.
[7] 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.
[8] 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.
[9] M.E. Kilmer, C.D. Martin, Factorization strategies for third-order tensors, Linear Algebra Appl. 435 (3) (2011) 641–658.
[10] Z. Zhang, S. Aeron, Exact tensor completion using t-SVD, IEEE Trans. Signal Process. 65 (6) (2017) 1511–1526.
[11] 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.
[12] D. Qiu, B. Yang, X. Zhang, Robust tensor completion via dictionary learning and generalized nonconvex regularization for visual data recovery, IEEE Trans. Circuits Syst. Video Technol. 34 (11) (2024) 11026–11039.
[13] Y. Qiu, G. Zhou, A. Wang, Q. Zhao, S. Xie, Balanced unfolding induced tensor nuclear norms for high-order tensor completion, IEEE Trans. Neural Netw. Learn. Syst. (2024). https://doi.org/10.1109/TNNLS.2024.3373384
[14] J.-B. Huang, A. Singh, N. Ahuja, Single image super-resolution from transformed self-exemplars, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 5197–5206.
[15] Y. Sun, L. Lei, X. Li, H. Sun, G. Kuang, Nonlocal patch similarity based heterogeneous remote sensing change detection, Pattern Recognit. 109 (2021) 107598.
[16] S.V. Venkatakrishnan, C.A. Bouman, B. Wohlberg, Plug-and-play priors for model based reconstruction, in: 2013 IEEE Global Conference on Signal and Information Processing, 2013, pp. 945–948.
[17] X.-L. Zhao, W.-H. Xu, T.-X. Jiang, Y. Wang, M.K. Ng, Deep plug-and-play prior for low-rank tensor completion, Neurocomputing 400 (2020) 137–149.
[18] K. Zhang, Y. Li, W. Zuo, L. Zhang, L. Van Gool, R. Timofte, Plug-and-play image restoration with deep denoiser prior, IEEE Trans. Pattern Anal. Mach. Intell. 44 (10) (2022) 6360–6376.
[19] Y. Hu, T.-X. Jiang, X.-L. Zhao, Degradation accordant plug-and-play for low-rank tensor completion, in: Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, 2022, pp. 1843–1849.
[20] Y. Qiu, G. Zhou, Q. Zhao, S. Xie, Noisy tensor completion via low-rank tensor ring, IEEE Trans. Neural Netw. Learn. Syst. 35 (1) (2024) 1127–1141. https://doi.org/10.1109/TNNLS.2022.3181378
[21] X.-L. Zhao, J.-H. Yang, T.-H. Ma, T.-X. Jiang, M.K. Ng, T.-Z. Huang, Tensor completion via complementary global, local, and nonlocal priors, IEEE Trans. Image Process. 31 (2021) 984–999.
[22] T.G. Kolda, B.W. Bader, Tensor decompositions and applications, SIAM Rev. 51 (3) (2009) 455–500.
[23] 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.
[24] X. Tong, H. Zhu, L. Cheng, Y.-C. Wu, Revisiting trace norm minimization for tensor tucker completion: a direct multilinear rank learning approach, Pattern Recognit. 158 (2025) 110995.
[25] 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.
[26] I.V. Oseledets, Tensor-train decomposition, SIAM J. Sci. Comput. 33 (5) (2011) 2295–2317.
[27] X. Tian, K. Xie, J. Wen, Tensor wheel completion with parallel matrix factorization and group smoothness for hyperspectral image recovery, Pattern Recognit. (2025) 112508.
[28] Y.-B. Zheng, T.-Z. Huang, X.-L. Zhao, Q. Zhao, T.-X. Jiang, Fully-connected tensor network decomposition and its application to higher-order tensor completion, in: Proceedings of the AAAI Conference on Artificial Intelligence, 35, 2021, pp. 11071–11078.
[29] 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.
[30] E. Kernfeld, M. Kilmer, S. Aeron, Tensor–tensor products with invertible linear transforms, Linear Algebra Appl. 485 (2015) 545–570.
[31] 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.
[32] 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.
[33] C. Liu, X. Wang, Y. Fan, S. Li, X. Qian, Decoupling degradations with recurrent network for video restoration in under-display camera, in: Proceedings of the AAAI Conference on Artificial Intelligence, 38, 2024, pp. 3558–3566.
[34] X. Xue, J. Yuan, T. Ma, L. Ma, Q. Jia, J. Zhou, Y. Wang, Degradation-decoupled and semantic-aggregated cross-space fusion for underwater image enhancement, Inf. Fus. 118 (2025) 102927.
[35] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, et al., Distributed optimization and statistical learning via the alternating direction method of multipliers, Found. Trends® Mach. Learning 3 (1) (2011) 1–122.
[36] E. Ryu, J. Liu, S. Wang, X. Chen, Z. Wang, W. Yin, Plug-and-play methods provably converge with properly trained denoisers, in: Proceedings of the International Conference on Machine Learning, 2019, pp. 5546–5557.
[37] Z.-Y. Wang, H.C. So, A.M. Zoubir, Low-rank tensor completion via novel sparsity-inducing regularizers, IEEE Trans. Signal Process. 72 (2024) 3519–3534.
[38] W. Xue, L. Zhang, X. Mou, A.C. Bovik, Gradient magnitude similarity deviation: a highly efficient perceptual image quality index, IEEE Trans. Image Process. 23 (2) (2014) 684–695. https://doi.org/10.1109/TIP.2013.2293423
[39] C. Xie, S. Liu, C. Li, M.-M. Cheng, W. Zuo, X. Liu, S. Wen, E. Ding, Image inpainting with learnable bidirectional attention maps, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 8858–8867.
[40] E. Candes, B. Recht, Exact matrix completion via convex optimization, Commun. ACM 55 (6) (2012) 111–119.