



Generalized nonconvex regularization for tensor RPCA and its applications in visual inpainting

Feng Zhang¹ · Hailin Wang² · Wenjin Qin¹ · Xile Zhao³ · Jianjun Wang¹

Accepted: 28 May 2023 / Published online: 5 July 2023

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract

As a demonstrated and foremost approach of extracting the key features from corrupted observations, tensor robust principal component analysis has been considered in various fields of data processing related to tensors. It is usually modeled as a low-rank and sparse tensor decomposition problem and can be solved by minimizing a simple convex program. However, convex optimization methods often fail to deeply explore the rank and sparsity of tensors, which leads to suboptimality. Based on tensor singular value decomposition, in this work, we introduce generalized nonconvex regularizers accommodating most popular nonconvex (and possibly nonsmooth) surrogate functions to be used as effective approximations of the tensor rank function and ℓ_0 -norm. The established unified frame equips universality for a large group of nonconvex surrogate functions. Moreover, we consider tube-wise sparse noise in addition to entry-wise sparse noise, which provides a better way of handling structured corrupted observations arising from practical issues. We further develop an efficient algorithm with convergence guarantees to implement generalized nonconvex optimization based upon the alternating direction method of multipliers. The satisfactory performance results of the proposed method are verified by simulations and visual inpainting applications.

Keywords Tensor robust principal component analysis · Tensor singular value decomposition · Low-rank tensor recovery · Group sparse · Nonconvex optimization

1 Introduction

With the rapid development of modern technologies, in most application scenarios such as hyperspectral image processing

✉ Jianjun Wang
wj@swu.edu.cn

Feng Zhang
zfm@swu.edu.cn

Hailin Wang
wanghailin97@163.com

Wenjin Qin
qinwenjin2021@163.com

Xile Zhao
xlzhao122003@163.com

¹ School of Mathematics and Statistics, Southwest University, Chongqing 400715, China

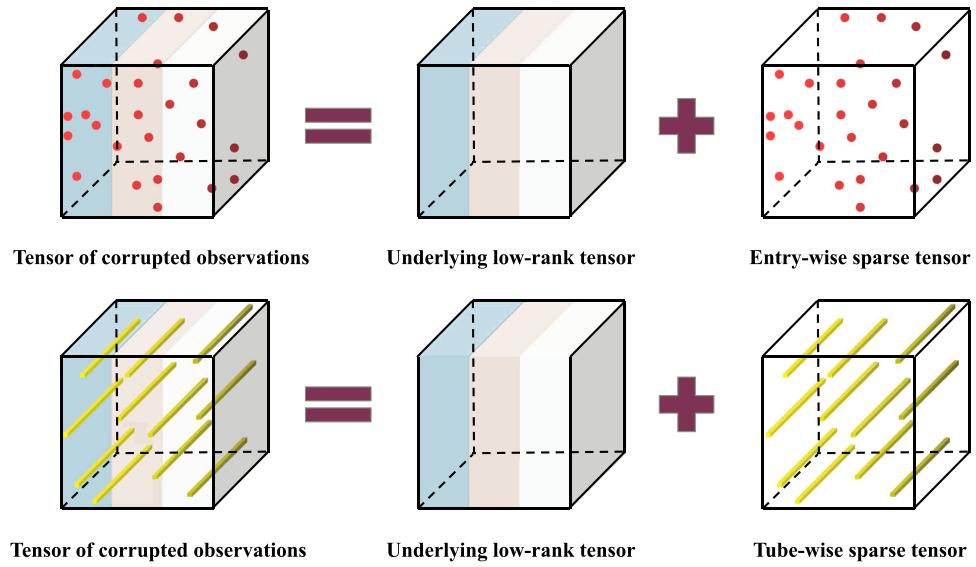
² School of Mathematics and Statistics, Xi'an JiaoTong University, Xi'an 710049, China

³ School of Mathematical Sciences/Research Center for Image and Vision Computing, University of Electronic Science and Technology of China, Chengdu 611731, China

[1, 2], facial recognition [3], and recommender systems [4, 5], tensors, or high-order arrays, have increasingly displaced low-order vectors and matrices as the object of interest. In recent years, great progress has been made in paradigms related to tensor analysis, such as tensor robust principal component analysis (tensor RPCA) [6], tensor completion [7, 8], and tensor compression [9]. As a modification of the widely used statistical procedure of classical PCA, tensor RPCA works well with respect to highly corrupted measurements, which has drawn significant attention in the field of tensor processing [10–14].

Formally, for a given data tensor $\mathcal{D} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ (this work focuses on the 3-way tensor), tensor RPCA seeks to extract the intrinsic low-rank structure from tensor data corrupted by grossly sparse noise: $\mathcal{D} = \mathcal{L}_0 + \mathcal{E}_0$, where \mathcal{L}_0 has low rank and \mathcal{E}_0 is (entry-wise) sparse with unknown support and arbitrary magnitude. See Fig. 1 (upper row) for an intuitive illustration. Many real applications fall into this pattern, e.g., background modeling [15], subspace clustering [16], and video compressive sensing [17]. This decomposition in low-rank and sparse tensors can be achieved by the following particular formulation of tensor

Fig. 1 Illustrations of tensor RPCA. Upper row: low-rank plus entry-wise sparse tensor decomposition from noisy tensor observations. Bottom row: low-rank plus tube-wise sparse tensor decomposition from noisy tensor observations



RPCA:

$$\min_{\mathcal{L}, \mathcal{E}} \text{rank}(\mathcal{L}) + \lambda \|\mathcal{E}\|_0, \quad \text{s.t. } \mathcal{D} = \mathcal{L} + \mathcal{E}, \quad (1)$$

where $\|\mathcal{E}\|_0 = \#\{\mathcal{E}_{ijk} \neq 0\}$ indicates ℓ_0 -norm which measures the sparsity of noise tensor \mathcal{E} ; $\text{rank}(\mathcal{L})$ denotes the rank of clean tensor \mathcal{L} and is used to measure its high-order sparsity, which has different notions corresponding to the different choice of tensor decompositions; $\lambda > 0$ is devoted to balancing these two nonconvex regularization terms. The formula (1) is a typical combinatorial optimization problem, and the straightforward approach to solve it is completely unpractical.

Instead of solving such an NP hard problem (1), the researchers minimize the convex relaxation of ℓ_0 -norm and tensor rank function:

$$\min_{\mathcal{L}, \mathcal{E}} f_1(\mathcal{L}) + \lambda f_2(\mathcal{E}), \quad \text{s.t. } \mathcal{D} = \mathcal{L} + \mathcal{E}, \quad (2)$$

where $f_1(\cdot)$ and $f_2(\cdot)$ denote two suitable convex surrogate functions, which would be a promising approach. For the low-rank part $f_1(\cdot)$, the choice of convex surrogates of $\text{rank}(\mathcal{L})$ depends on tensor decompositions and associated rank definitions. The most popular tensor ranks are CP rank induced by CANDECOMP/PARAFAC (CP) decomposition [18], Tucker rank induced by Tucker decomposition [19], tensor train (TT) rank induced by TT decomposition [20], and tensor tubal rank induced by tensor singular value decomposition (t-SVD) [21]. In 2013, based on Tucker decomposition, the sum of nuclear norms (SNN) was proposed by Liu et al. [22] and used as the convex surrogate of Tucker rank. In 2014, Zhang et al. [23] defined the tensor nuclear norm (TNN) within the t-SVD framework as the convex surrogate of tensor multi rank, which was further rigorously redefined

by Lu et al. [6, 24]. In 2017, based on TT decomposition, Bengua et al. [25] provided the TT nuclear norm (TTNN) as the convex surrogate of the TT rank. In 2018, Friedland et al. [26] introduced the tensor nuclear norm based on CP decomposition (cTNN) as the convex surrogate of CP rank. In addition, for the sparse part $f_2(\cdot)$, most of the methods use the ℓ_1 -norm, i.e., $\|\mathcal{E}\|_1 = \sum_{ijk} |\mathcal{E}_{ijk}|$, as the convex surrogate of the ℓ_0 -norm. Then, the researchers apply the abovementioned convex surrogates to the model (2), and the resulting convex tensor RPCA problem can be well solved [6, 23, 24, 27].

It is well known that the use of the ℓ_1 -norm in sparse recovery has become widespread. This is mainly due to its concise form and convexity, which makes it easy to solve and comes with attractive theoretical properties. However, several reports have shown that the ℓ_1 -norm is biased and suboptimal since it is a loose approximation of the ℓ_0 -norm and overpenalizes large coefficients of sparse components [28]. Thus, selecting ℓ_1 -norm for $f_2(\cdot)$ is not an optimal strategy for measuring the sparsity of tensor \mathcal{E} . As previously mentioned, $\text{rank}(\cdot)$ acts as ℓ_0 -norm on the vector of tensor singular values. Moreover, the convex surrogate function $f_1(\cdot)$ of $\text{rank}(\cdot)$, covering various tensor nuclear norms, is essentially the ℓ_1 -norm on the vector of tensor singular values and achieves a low rank by encouraging high-order sparsity on the tensor singular values. Similarly, the obtained solution by optimization solvers with $f_1(\cdot)$ is usually suboptimal since $f_1(\cdot)$ is a loose approximation of the rank function of tensor \mathcal{L} and larger tensor singular values are penalized more heavily in the $f_1(\cdot)$ than smaller ones. To address this imbalance, there are a variety of nonconvex penalties available to better approximate the ℓ_0 -norm/rank function and enhance (high-order) sparsity. These include exponential-type penalty (ETP) [29] and Geman [30], Laplace [31], logarithm [32], ℓ_p -

Table 1 A list of various nonconvex surrogate functions and their supergradients [42]

Penalty	Formula $g(x)$, $x \geq 0$, $\lambda > 0$, $\gamma > 1$	Supergradient $\partial g(x)$
ETP	$\frac{\lambda}{1-\exp(-\gamma)}(1 - \exp(-\gamma x))$	$\frac{\lambda\gamma}{1-\exp(-\gamma)} \exp(-\gamma x)$
Geman	$\frac{\lambda x}{x+\gamma}$	$\frac{\lambda\gamma}{(x+\gamma)^2}$
Laplace	$\lambda(1 - \exp(-\frac{x}{\gamma}))$	$\frac{\lambda}{\gamma} \exp(-\frac{x}{\gamma})$
Logarithm	$\frac{\lambda}{\log(\gamma+1)} \log(\gamma x + 1)$	$\frac{\gamma\lambda}{(\gamma x+1) \log(\gamma+1)}$
ℓ_p	λx^p	$\begin{cases} +\infty, & \text{if } x = 0, \\ \lambda p x^{p-1}, & \text{if } x > 0. \end{cases}$
MCP	$\begin{cases} \lambda x - \frac{x^2}{2\gamma}, & \text{if } x < \gamma\lambda, \\ \frac{1}{2}\gamma\lambda^2, & \text{if } x \geq \gamma\lambda. \end{cases}$	$\begin{cases} \lambda - \frac{x}{\gamma}, & \text{if } x < \gamma\lambda, \\ 0, & \text{if } x \geq \gamma\lambda. \end{cases}$
Capped-L1	$\begin{cases} \lambda x, & \text{if } x < \gamma, \\ \gamma\lambda, & \text{if } x \geq \gamma. \end{cases}$	$\begin{cases} \lambda, & \text{if } x < \gamma, \\ , & \text{if } x = \gamma, \\ 0, & \text{if } x > \gamma. \end{cases}$
SCAD	$\begin{cases} \lambda x, & \text{if } x \leq \lambda, \\ \frac{-x^2+2\lambda\gamma x-\lambda^2}{2(\gamma-1)}, & \text{if } \lambda < x \leq \gamma\lambda, \\ \frac{\lambda^2(\gamma+1)}{2}, & \text{if } x > \gamma\lambda. \end{cases}$	$\begin{cases} \lambda, & \text{if } x \leq \lambda, \\ \frac{\gamma\lambda-x}{\gamma-1}, & \text{if } \lambda < x \leq \gamma\lambda, \\ 0, & \text{if } x > \gamma\lambda. \end{cases}$

norm ($0 < p < 1$) [33], minimax concave penalty (MCP) [34], Capped-L1 [35] and smoothly clipped absolute deviation (SCAD) penalties [36]. The definitions of each are summarized in Table 1. Equipped with the above nonconvex surrogates, some optimization programs have demonstrated excellent performance in the tensor RPCA problem [37–41].

Inspired by previous research, in this paper, we introduce a generalized nonconvex optimization method for tensor RPCA with SCAD and ℓ_p -norm ($0 < p < 1$) approaches as special cases within the framework of t-SVD induced by the tensor-tensor product [21]. The primary reason for t-SVD adoption is that it is a more natural extension of SVD than other tensor decomposition forms and involves a consistent set of mathematical operations with the matrix cases (see Section 2). The proposed two generalized nonconvex regularizers accommodate all the nonconvex surrogate functions mentioned above, which can achieve a better approximation of the tubal rank function and ℓ_0 -norm of tensors than the convex surrogates $f_1(\cdot)$ and $f_2(\cdot)$ respectively. We develop an efficient algorithm based on the standard alternating direction method of multipliers (ADMM) [43] to solve the tensor RPCA program with two generalized nonconvex regularizers. The algorithm architecture relies on solving two kinds of tensor subproblems: weighted tensor nuclear norm minimization and weighted tensor ℓ_1 -norm minimization. As a result of the nonnegative and monotonically decreasing of the supergradients of the nonconvex functions summarized in Table 1, two closed-form solutions are provided by the weighted tensor singular value threshold and the weighted tensor soft threshold.

As far as we know, recent studies [6, 13, 44] on the tensor RPCA problem usually assume that noise is entry-wise sparse (uniformly distributed across the tensor \mathcal{E}_0 , see Fig. 1 (upper row)) and thus cannot effectively handle structured corrup-

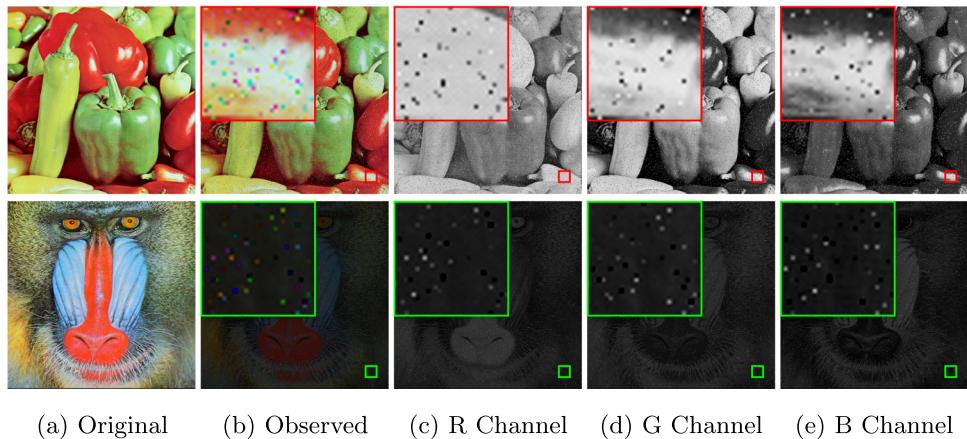
tions. At present, there are two popular classes of structured noise tensors of interest, namely, slice-wise sparse tensors [10] and tube-wise sparse tensors [23, 45]. The focus of this work is on the latter. Tube-wise sparsity can be defined as the number of nonzero tubes along the third dimension of a tensor. It may occur when the pixel values of a color image are corrupted. Figure 2 presents a comparison of entry-wise and tube-wise corruption of color images¹. As shown in the Fig. 2, the “PeppersRGB” image corrupted by entry-wise noise has inconsistent noise location distributions in the R, G, and B channels, and the tube-wise noise-corrupted “BaboonRGB” image is the opposite of the above. It is very important to consider the latter when actually inpainting an image. Additionally, this latter problem poses a greater challenge than corruptions on 3 channels at different positions. Therefore, in this work, we will further extend the generalized nonconvex optimization method proposed above to the case of low-tubal-rank plus tube-wise sparse tensor decomposition. See Fig. 1 (bottom row) for an intuitive illustration.

The contributions of this work are summarized as follows:

1. In the framework of t-SVD, we developed a more effective way to solve the tensor RPCA problem via generalized nonconvex regularization. Most popular nonconvex surrogate functions can be contained by the proposed two generalized nonconvex regularizers, which are better able to approximate the tubal rank function and ℓ_0 -norm of tensors than their convex counterparts respectively.
2. For the sparse component, in addition to entry-wise noise, we also consider another kind of structured corruption, namely, tube-wise noise, which arises frequently in color image inpainting applications. In addition, the advan-

¹ Originated from <http://www.eecs.qmul.ac.uk/~phao/IP/Images/>

Fig. 2 Diagrams of noise location distributions of corrupted RGB images with a 10x magnification of a partial area. (a) shows the clean data from top to bottom entitled “PeppersRGB” and “BaboonRGB”. (b) shows the observed “PeppersRGB” and “BaboonRGB” images by corrupted entry-wise noise and tube-wise noise, respectively. (c)-(e) show noise location distributions in the R, G, and B channels of corrupted RGB images



tages of the proposed optimization methods over others that ignore this structure are also verified.

3. We propose an efficient algorithm with two closed-form solutions that is based on the standard alternating direction method of multiplier (ADMM) to solve the nonconvex tensor RPCA problem. A theoretical guarantee for the convergence of proposed algorithm is provided. Extensive experiments on simulated tensor data decomposition and visual inpainting tasks reveal that the proposed method is superior to several existing tensor RPCA methods in terms of performance.

There are several typical related efforts that should be highlighted here. As far as the authors are aware, Lu et al. first introduced generalized nonconvex regularization into the low-rank matrix minimization model in [42, 46, 47]. Since then, there have been many relevant studies with similar ideas emerging in matrix inverse problems, including background modeling and image restoration [48–51]. While the aforementioned techniques may encounter difficulties with the multilinear structure of tensor data, our approach excels in addressing this issue.

In 2021, Wang et al. [52] extended generalized nonconvex regularization to the case of third-order tensors but only for tensor completion problems. It is apparent that solving the tensor RPCA problem with noisy observations, as is proposed in this paper, where the magnitude and support of the noise are unknown, poses a greater challenge compared to completing a tensor without significant noise. Based on the tensor RPCA paradigm, Chen et al. [39] successfully applied this strategy to the problem of multiview subspace clustering but did not introduce a generalized nonconvex surrogate for the sparse component, whereas we have done so. Recently, Zhang et al. [53] proposed the weighted tensor average rank combined with generalized nonconvex regularization as an effective solution to the tensor RPCA problem. However, this approach does not consider the structural information of the sparse residue, which is a common occurrence in practical applications. In contrast, our approach takes this into account.

The remainder of the paper is organized as follows. In Section 2, we introduce the necessary notations and preliminaries for the sections that follow. In Section 3, we introduce the formulation of the problem and two key proximity operators. Section 4 is devoted to an efficient algorithm with convergence guarantees for solving the proposed nonconvex tensor RPCA problem. Section 5 contains a numerical study evaluating the performance of the proposed algorithm. A summary and suggestions for future work are presented in Section 6.

2 Notations and preliminaries

2.1 Notations

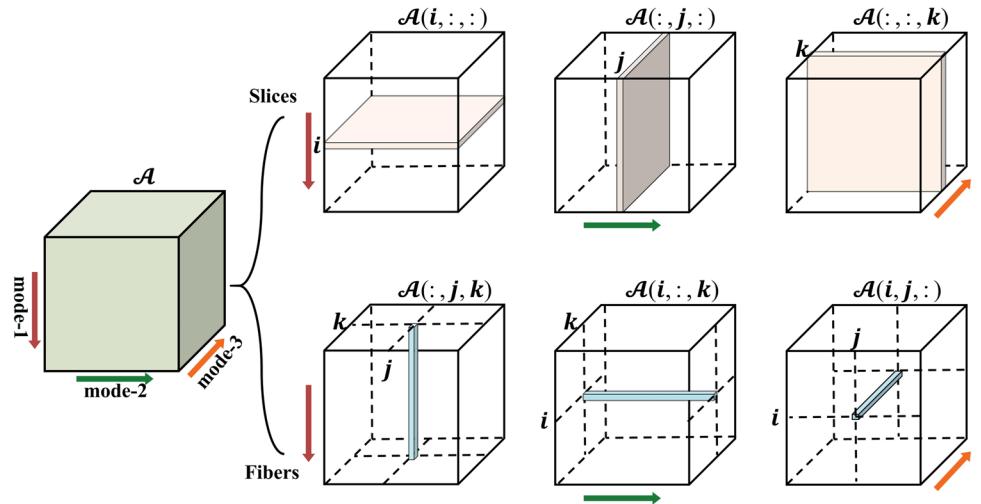
For brevity, we summarize the main notations in Table 2. The specific meanings and calculations of some of these notations are introduced at their initial appearance in the paper. Specifically, we demonstrate in Fig. 3 three different slices and three types of fibers that constitute a third-order tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ to aid readers in better understanding the subsequent operations on the tensor.

For a tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, we use $\bar{\mathcal{A}}$ to represent the result of discrete Fourier transformation of \mathcal{A} along the third dimension. The fast Fourier transform (FFT) is a more prevalent technique used in practice. MATLAB’s command can be used to rewrite the definition: $\bar{\mathcal{A}} = \text{fft}(\mathcal{A}, [], 3)$ and ifft is defined as the reverse operation, i.e., $\mathcal{A} = \text{ifft}(\bar{\mathcal{A}}, [], 3)$. Let $\bar{\mathcal{A}} \in \mathbb{C}^{n_1 n_2 \times n_3}$ be the block diagonal matrix with each block on the diagonal as the frontal slice $\bar{\mathcal{A}}^{(i)}$ of $\bar{\mathcal{A}}$, i.e.,

$$\bar{\mathcal{A}} = \text{bdiag}(\bar{\mathcal{A}}) = \begin{bmatrix} \bar{\mathcal{A}}^{(1)} & & & \\ & \bar{\mathcal{A}}^{(2)} & & \\ & & \ddots & \\ & & & \bar{\mathcal{A}}^{(n_3)} \end{bmatrix}. \quad (3)$$

Table 2 Summary of main notations in the paper

Notations	Descriptions	Notations	Descriptions
a	A scalar.	\mathbf{A}	A matrix.
\mathbf{a}	A vector.	\mathcal{A}	A tensor.
\mathcal{A}_{ijk} or a_{ijk}	The (i, j, k) -th entry of \mathcal{A} .	$\ \mathbf{a}\ _2 = \sqrt{\sum_i a_i ^2}$	The ℓ_2 -norm of \mathbf{a} .
$\mathcal{A}(i, :, :)$	The i -th horizontal slice of \mathcal{A} .	$\ \mathbf{A}\ _F = \sqrt{\sum_{i,j} a_{ij} ^2}$	The Frobenius norm of \mathbf{A} .
$\mathcal{A}(:, j, :)$	The j -th lateral slice of \mathcal{A} .	$\ \mathcal{A}\ _1 = \sum_{i,j,k} a_{ijk} $	The ℓ_1 -norm of \mathcal{A} .
$\mathcal{A}(:, :, k)$ or $\mathbf{A}^{(k)}$	The k -th frontal slice of \mathcal{A} .	$\ \mathcal{A}\ _\infty = \max_{i,j,k} a_{ijk} $	The infinity norm of \mathcal{A} .
$\mathcal{A}(i, j, :)$	The (i, j) -th tubal fiber of \mathcal{A} .	$\ \mathcal{A}\ _F = \sqrt{\sum_{i,j,k} a_{ijk} ^2}$	The Frobenius norm of \mathcal{A} .
$\tilde{\mathcal{A}}$	The DFT of \mathcal{A} .	$\ \mathcal{A}\ _0 = \#\{\mathcal{A}_{ijk} \neq 0\}$	The ℓ_0 -norm of \mathcal{A} .
$\tilde{a}_{ij}(\mathcal{A})$	The i -th singular value of $\tilde{\mathcal{A}}(:, :, j)$.	$\langle \mathcal{A}, \mathbf{B} \rangle = \sum_k \langle \mathbf{A}^{(k)}, \mathbf{B}^{(k)} \rangle$	The inner product of \mathcal{A} and \mathbf{B} .
\mathcal{A}^\top	The conjugate transpose of \mathcal{A} .	$\ \mathcal{A}\ _{\circledast}$	The tensor nuclear norm of \mathcal{A} .
$\mathcal{A}^{(t)}$	The t -th iteration of \mathcal{A} .	$\ \mathcal{A}\ _{w,\circledast}$	The weighted tensor nuclear norm of \mathcal{A} .
$\hat{\mathcal{A}}$	The optimal solution of \mathcal{A} .	$\ \mathcal{A}\ _{\dot{w},1}$	The weighted tensor ℓ_1 -norm of entry-wise sparse \mathcal{A} .
$\bar{\mathcal{A}}$ or $\text{bdiag}(\tilde{\mathcal{A}})$	The block diagonal matrix.	$\ \mathcal{A}\ _{\ddot{w},1}$	The weighted tensor ℓ_1 -norm of tube-wise sparse \mathcal{A} .
$\text{bcirc}(\mathcal{A})$	The block circulant matrix.	$\text{rank}_t(\mathcal{A})$	The tensor tubal rank of \mathcal{A} .
$\text{unfold}(\mathcal{A})$ and $\text{fold}(\mathbf{B})$	The tensor unfold operator and its inverse operator.	$n_{(1)}$ and $n_{(2)}$	$n_{(1)} = \max\{n_1, n_2\}$ and $n_{(2)} = \min\{n_1, n_2\}$.
$\tilde{g}_1(\mathcal{A})$	The generalized nonconvex surrogate function of $\text{rank}_t(\mathcal{A})$.	$g_1(\cdot)$ and $g_2(\cdot)$	The generalized nonconvex surrogate functions shown in Table 1.
$\tilde{g}_2(\mathcal{A})$	The generalized nonconvex surrogate function of $\ \mathcal{A}\ _0$.	w_{ij}, \dot{w}_{ijk} and \ddot{w}_{ijk}	The i -th, ijk -th and ij -th values of the weight parameters \mathbf{W} , $\dot{\mathbf{W}}$ and $\ddot{\mathbf{W}}$, respectively.

Fig. 3 Illustrations of slices and fibers of a third-order tensor

The block circulant matrix $\text{bcirc}(\mathcal{A}) \in \mathbb{R}^{n_1 n_3 \times n_2 n_3}$ of \mathcal{A} is defined as follows:

$$\text{bcirc}(\mathcal{A}) = \begin{bmatrix} \mathcal{A}^{(1)} & \mathcal{A}^{(n_3)} & \dots & \mathcal{A}^{(2)} \\ \mathcal{A}^{(2)} & \mathcal{A}^{(1)} & \dots & \mathcal{A}^{(3)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{A}^{(n_3)} & \mathcal{A}^{(n_3-1)} & \dots & \mathcal{A}^{(1)} \end{bmatrix}. \quad (4)$$

Remark 1 For a tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, we have $\|\mathcal{A}\|_F = \frac{1}{\sqrt{n_3}} \|\bar{\mathcal{A}}\|_F = \frac{1}{\sqrt{n_3}} \|\text{bdiag}(\bar{\mathcal{A}})\|_F$, which is used in the convergence proof of the algorithm. It can be proved by the relation between the block circulant matrix and block diagonal matrix: $(\mathbf{F}_{n_3} \otimes \mathbf{I}_{n_1}) \cdot \text{bcirc}(\mathcal{A}) \cdot (\mathbf{F}_{n_3}^{-1} \otimes \mathbf{I}_{n_2}) = \bar{\mathcal{A}}$, and the unitary invariance of the Frobenius norm, where \otimes denotes the Kronecker product, \mathbf{F}_{n_3} is a discrete Fourier transformation matrix, \mathbf{I}_{n_1} (or \mathbf{I}_{n_2}) is an $n_1 \times n_1$ (or $n_2 \times n_2$) identity matrix, $(\mathbf{F}_{n_3} \otimes \mathbf{I}_{n_1})/\sqrt{n_3}$ is unitary.

The `unfold` operator which can map $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ to a matrix of size $n_1 n_3 \times n_2$ and its inverse operator `fold`, respectively, are defined as

$$\text{unfold}(\mathcal{A}) = \begin{bmatrix} \mathcal{A}^{(1)} \\ \mathcal{A}^{(2)} \\ \vdots \\ \mathcal{A}^{(n_3)} \end{bmatrix} \quad \text{and} \quad \text{fold}(\text{unfold}(\mathcal{A})) = \mathcal{A}. \quad (5)$$

2.2 Preliminaries

This subsection presents the definition of the t-product, which enables us to calculate the product of third-order tensors and perform associated algebraic operations including computing the t-SVD of tensors, the tensor tubal rank, and the tensor nuclear norm.

Definition 1 (t-product [21]) Let $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{B} \in \mathbb{R}^{n_2 \times n_4 \times n_3}$. Then the t-product $\mathcal{A} * \mathcal{B}$ is defined as a tensor of size $n_1 \times n_4 \times n_3$ by

$$\mathcal{A} * \mathcal{B} = \text{fold}(\text{bcirc}(\mathcal{A}) \cdot \text{unfold}(\mathcal{B})). \quad (6)$$

Algorithm 1 proposes an efficient method for computing the t-product using FFT instead of relying on the above formula (6). This is because the t-product is equivalent to matrix multiplication in the Fourier transform domain.

Next, we review the concepts of conjugate transposition and identity tensor which will be used for the remainder of this paper.

Definition 2 (Conjugate transpose [21]) Let $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$. Then its conjugate transpose \mathcal{A}^\top is an $n_2 \times n_1 \times n_3$ tensor obtained by conjugate transposing every frontal slice of

Algorithm 1 Computing the t-product of third-order tensors [21].

Input: $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{B} \in \mathbb{R}^{n_2 \times n_4 \times n_3}$.

Output: $\mathcal{C} = \mathcal{A} * \mathcal{B} \in \mathbb{R}^{n_1 \times n_4 \times n_3}$.

1: $\bar{\mathcal{A}} = \text{fft}(\mathcal{A}, [], 3)$ and $\bar{\mathcal{B}} = \text{fft}(\mathcal{B}, [], 3)$.

2: **for** $k = 1, \dots, n_3$ **do**

3: $\bar{\mathcal{C}}^{(k)} = \bar{\mathcal{A}}^{(k)} \bar{\mathcal{B}}^{(k)}$.

4: **end for**

5: $\mathcal{C} = \text{ifft}(\bar{\mathcal{C}}, [], 3)$.

\mathcal{A} and then reversing the order of the conjugate transposed frontal slice 2 through n_3 .

Definition 3 (Identity tensor [21]) The identity tensor $\mathcal{I} \in \mathbb{R}^{n \times n \times n_3}$ is the tensor whose first frontal slice is the identity matrix \mathbf{I}_n and other frontal slices are all zeros.

The concept of orthogonal tensors becomes straightforward with the establishment of conjugate transpose and identity operators.

Definition 4 (Orthogonal tensor [21]) A tensor $\mathcal{Q} \in \mathbb{R}^{n \times n \times n_3}$ is orthogonal if it satisfies

$$\mathcal{Q}^\top * \mathcal{Q} = \mathcal{Q} * \mathcal{Q}^\top = \mathcal{I}. \quad (7)$$

There also exists the notion of an F-diagonal tensor.

Definition 5 (F-diagonal tensor [21]) A tensor is called an F-diagonal if each of its frontal slices is a diagonal matrix.

Building on the operations of tensors introduced above, we can now describe the t-SVD of a third-order tensor, as well as the tensor tubal rank and tensor nuclear norm induced by it.

Theorem 1 (t-SVD [21]) Let $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$. Then, it can be decomposed as

$$\mathcal{A} = \mathcal{U}_\mathcal{A} * \mathcal{S}_\mathcal{A} * \mathcal{V}_\mathcal{A}^\top, \quad (8)$$

where $\mathcal{U}_\mathcal{A} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$, $\mathcal{V}_\mathcal{A} \in \mathbb{R}^{n_2 \times n_2 \times n_3}$ are orthogonal, and $\mathcal{S}_\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is an F-diagonal tensor. See Fig. 4 for a visual representation.

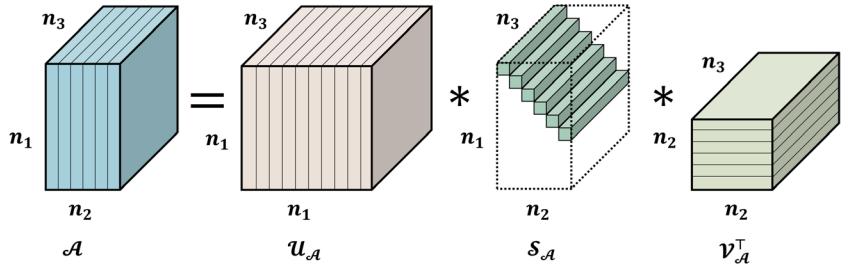
Algorithm 2 shows that the t-SVD can be efficiently computed by performing matrix SVD in the Fourier transform domain.

Next, we introduce the notion of tensor tubal rank. It has been demonstrated that most data in practice have a low-tubal-rank structure, which means that they can be well approximated by low-tubal-rank tensors.

Definition 6 (Tensor tubal rank [23, 54]) For $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, its t-SVD is $\mathcal{A} = \mathcal{U}_\mathcal{A} * \mathcal{S}_\mathcal{A} * \mathcal{V}_\mathcal{A}^\top$. The tensor tubal rank of \mathcal{A} is defined as the number of nonzero singular tubes of the F-diagonal tensor $\mathcal{S}_\mathcal{A}$, i.e.,

$$\text{rank}_t(\mathcal{A}) = \#\{i : \mathcal{S}_\mathcal{A}(i, i, :) \neq \mathbf{0}\}. \quad (9)$$

Fig. 4 An illustration of the t-SVD of a third-order tensor



Algorithm 2 Computing the t-SVD of third-order tensors [21].

Input: $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{B} \in \mathbb{R}^{n_2 \times n_4 \times n_3}$.
Output: $\mathcal{U}_{\mathcal{A}} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$, $\mathcal{S}_{\mathcal{A}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, and $\mathcal{V}_{\mathcal{A}} \in \mathbb{R}^{n_2 \times n_2 \times n_3}$.

- 1: $\tilde{\mathcal{A}} = \text{ffft}(\mathcal{A}, [], 3)$.
- 2: **for** $k = 1, \dots, n_3$ **do**
- 3: $[\tilde{\mathcal{U}}_{\mathcal{A}}^{(k)}, \tilde{\mathcal{S}}_{\mathcal{A}}^{(k)}, \tilde{\mathcal{V}}_{\mathcal{A}}^{(k)}] = \text{SVD}(\tilde{\mathcal{A}}^{(k)})$.
- 4: **end for**
- 5: $\mathcal{U}_{\mathcal{A}} = \text{ifft}(\tilde{\mathcal{U}}_{\mathcal{A}}, [], 3)$, $\mathcal{S}_{\mathcal{A}} = \text{ifft}(\tilde{\mathcal{S}}_{\mathcal{A}}, [], 3)$, and $\mathcal{V}_{\mathcal{A}} = \text{ifft}(\tilde{\mathcal{V}}_{\mathcal{A}}, [], 3)$.

The definition of the tensor nuclear norm, which has been applied as the most popular convex relaxation of the tensor tubal rank, will be given.

Definition 7 (Tensor nuclear norm [6, 24]) For $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, its t-SVD is $\mathcal{A} = \mathcal{U}_{\mathcal{A}} * \mathcal{S}_{\mathcal{A}} * \mathcal{V}_{\mathcal{A}}^T$. The tensor nuclear norm is denoted as

$$\|\mathcal{A}\|_{\circledast} = \sum_{i=1}^{n_2} \mathcal{S}_{\mathcal{A}}(i, i, 1). \quad (10)$$

3 Tensor RPCA via generalized nonconvex regularization

In this section, we first define two generalized nonconvex surrogates for low-tubal-rank and sparse components to achieve better approximations of the tubal rank function and ℓ_0 -norm of tensors than the convex relaxation functions can achieve. In particular, we consider two types of grossly sparse noise: entry-wise and tube-wise sparse noise. Then, a nonconvex optimization model for tensor RPCA is proposed based on the generalized nonconvex surrogates introduced above. Finally, we develop two tensor threshold operators to be used later.

3.1 Formulation of problem

Based on the t-SVD induced by the tensor-tensor product, we now need to recover the tensor \mathcal{L} of low tubal rank by replacing $\text{rank}(\mathcal{L})$ in equation (1) with $\text{rank}_t(\mathcal{L})$. We know that the TNN redefined by Lu et al. [6, 24] is an excellent convex alternative to the nonconvex tubal rank function $\text{rank}_t(\mathcal{L})$, but it is sometimes biased and suboptimal in enhancing low

tubal rankness. Thus, next we give a new generalized nonconvex surrogate of $\text{rank}_t(\mathcal{L})$ to avoid this. Suppose that the low-tubal-rank tensor \mathcal{L} of $n_1 \times n_2 \times n_3$ has the t-SVD, i.e., $\mathcal{L} = \mathcal{U}_{\mathcal{L}} * \mathcal{S}_{\mathcal{L}} * \mathcal{V}_{\mathcal{L}}^T$. Using Definition 7 and the property induced by the inverse DFT, we have

$$\begin{aligned} \|\mathcal{L}\|_{\circledast} &= \sum_{i=1}^{n_2} \mathcal{S}_{\mathcal{L}}(i, i, 1) = \frac{1}{n_3} \sum_{i=1}^{n_2} \sum_{j=1}^{n_3} \bar{\mathcal{S}}_{\mathcal{L}}(i, i, j) \\ &= \frac{1}{n_3} \sum_{i=1}^{n_2} \sum_{j=1}^{n_3} \bar{\sigma}_{ij}(\mathcal{L}), \end{aligned} \quad (11)$$

where $\bar{\sigma}_{ij}(\mathcal{L})$ is defined as the i -th singular value of $\bar{\mathcal{L}}(:, :, j)$. Based on this, we define the following generalized nonconvex surrogate of $\text{rank}_t(\mathcal{L})$:

$$\tilde{g}_1(\mathcal{L}) := \frac{1}{n_3} \sum_{i=1}^{n_2} \sum_{j=1}^{n_3} g_1(\bar{\sigma}_{ij}(\mathcal{L})), \quad (12)$$

where $g_1 : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ is continuous, concave monotonically increasing on $[0, \infty)$ and possibly nonsmooth. It is not difficult to verify that all the nonconvex surrogates in Table 1 satisfy these properties.

For the sparse component \mathcal{E} , the ℓ_1 -norm

$$\|\mathcal{E}\|_1 = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \sum_{k=1}^{n_3} |\mathcal{E}_{ijk}|, \quad (13)$$

as the tightest convex relaxation of ℓ_0 -norm in equation (1), plays an important role in convex optimization. However, we know that it is sometimes biased and suboptimal in promoting sparsity. Therefore, we give the following generalized nonconvex surrogate of ℓ_0 -norm:

$$\tilde{g}_2(\mathcal{E}) := \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \sum_{k=1}^{n_3} g_2(h(|\mathcal{E}_{ijk}|)), \quad (14)$$

where g_2 has the same properties as g_1 . Here, we consider two types of grossly sparse noise. When the tensor \mathcal{E} is an

entry-wise sparse noise tensor, $h(\cdot) = \mathbf{1}$ is defined as an identity function. Then, we have

$$\tilde{g}_2(\mathcal{E}) = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \sum_{k=1}^{n_3} g_2(|\mathcal{E}_{ijk}|). \quad (15)$$

When the tensor \mathcal{E} is a tube-wise sparse noise tensor, $h(\cdot) = \|\cdot\|_2$ is defined as an ℓ_2 -norm. Then, we have

$$\tilde{g}_2(\mathcal{E}) = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} g_2(\|\mathcal{E}_{ij}\|_2). \quad (16)$$

Next, we propose the following generalized formulation for nonconvex regularized tensor RPCA:

$$\min_{\mathcal{L}, \mathcal{E}} \tilde{g}_1(\mathcal{L}) + \tilde{g}_2(\mathcal{E}) \quad \text{s.t. } \mathcal{D} = \mathcal{L} + \mathcal{E}, \quad (17)$$

where $\tilde{g}_1(\cdot)$ and $\tilde{g}_2(\cdot)$ are generalized nonconvex surrogates for low-tubal-rank and sparse inducing, respectively, as defined above. For simplicity, we always assume $\lambda = 1$ in $g_1(\cdot)$. When $g_1(\cdot) = \cdot$, $g_2(\cdot) = \lambda \cdot$ and $h(\cdot) = \mathbf{1}$, $\tilde{g}_1(\cdot)$ and $\tilde{g}_2(\cdot)$ reduce to $\|\cdot\|_{\otimes}$ and $\|\cdot\|_1$, respectively, and this formulation reduces to the tensor principal component pursuit program considered by Lu et al. in [6].

3.2 WTSVT and WTST operators

The tensor RPCA program (17) with two generalized nonconvex regularizers will be transformed into two subproblems, weighted tensor nuclear norm minimization and weighted tensor ℓ_1 -norm minimization, to be solved in Section 4.1. Here, we give in advance two key proximity operators for solving these two kinds of subproblems.

We first present the definition of the weighted tensor nuclear norm and then introduce the weighted tensor nuclear norm minimization problem and its solver.

Definition 8 (Weighted tensor nuclear norm [52]) For $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, its t-SVD is $\mathcal{A} = \mathcal{U}_{\mathcal{A}} * \mathcal{S}_{\mathcal{A}} * \mathcal{V}_{\mathcal{A}}^{\top}$. The weighted tensor nuclear norm is denoted as

$$\|\mathcal{A}\|_{w, \otimes} = \frac{1}{n_3} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} w_{ij} \bar{\sigma}_{ij}(\mathcal{A}), \quad (18)$$

where $\bar{\sigma}_{ij}(\mathcal{A}) := \bar{\mathcal{S}}_{\mathcal{A}}(i, i, j)$ and w_{ij} is the ij -th weight parameter.

Theorem 2 ([52]) For any $\tau > 0$, $\mathcal{B} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $0 \leq w_{1j} \leq w_{2j} \leq \dots \leq w_{n_{(2)}j}$, $j = 1, \dots, n_3$, a globally optimal solution to the following weighted tensor nuclear norm minimization problem

$$\min_{\mathcal{A}} \tau \|\mathcal{A}\|_{w, \otimes} + \frac{1}{2} \|\mathcal{A} - \mathcal{B}\|_F^2 \quad (19)$$

is given by the weighted tensor singular value thresholding (WTSVT)

$$\hat{\mathcal{A}} = \mathcal{U}_{\mathcal{B}} * \Phi_{w, \tau}(\mathcal{S}_{\mathcal{B}}) * \mathcal{V}_{\mathcal{B}}^{\top}, \quad (20)$$

where $\mathcal{B} = \mathcal{U}_{\mathcal{B}} * \mathcal{S}_{\mathcal{B}} * \mathcal{V}_{\mathcal{B}}^{\top}$ is the t-SVD of \mathcal{B} and $\Phi_{w, \tau}(\mathcal{S}_{\mathcal{B}}) = \text{ifft}((\bar{\mathcal{S}}_{\mathcal{B}} - \tau \mathcal{W})_+, [\cdot], 3)$, in which $\mathcal{W} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is f-diagonal whose diagonal entries of the j -th frontal slice are equal to the j -th column of the weight matrix $\mathbf{W} = (w_{ij})_{n_{(2)} \times n_3}$.

Proof Readers interested in the proof are encouraged to refer to [52]. \square

The detailed WTSVT calculation process is summarized in Algorithm 3.

Algorithm 3 WTSVT [52].

Input: $\mathcal{B} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, $\mathcal{W} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, and $\tau > 0$.

Output: $\hat{\mathcal{A}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$.

- 1: $\bar{\mathcal{B}} = \text{fft}(\mathcal{B}, [], 3)$.
 - 2: **for** $k = 1, \dots, n_3$ **do**
 - 3: $(\bar{\mathcal{U}}_{\mathcal{B}}^{(k)}, \bar{\mathcal{S}}_{\mathcal{B}}^{(k)}, \bar{\mathcal{V}}_{\mathcal{B}}^{(k)}) = \text{SVD}(\bar{\mathcal{B}}^{(k)})$;
 - 4: $\bar{\mathcal{A}}^{(k)} = \bar{\mathcal{U}}_{\mathcal{B}}^{(k)} \cdot (\bar{\mathcal{S}}_{\mathcal{B}}^{(k)} - \tau \mathcal{W}^{(k)})_+ \cdot (\bar{\mathcal{V}}_{\mathcal{B}}^{(k)})^{\top}$.
 - 5: **end for**
 - 6: $\hat{\mathcal{A}} = \text{ifft}(\bar{\mathcal{A}}, [], 3)$.
-

Remark 2 In [42], to handle the nonsmooth penalty $g(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}$, Lu et al. first define the supergradient $\mathbf{v} \in \mathbb{R}^n$ of concave $g(\cdot)$ at the nonsmooth point $\mathbf{x} \in \mathbb{R}^n$. That is, for every $\mathbf{y} \in \mathbb{R}^n$, \mathbf{v} satisfies the following inequality:

$$g(\mathbf{x}) + \langle \mathbf{v}, \mathbf{y} - \mathbf{x} \rangle \geq g(\mathbf{y}), \quad (21)$$

and the set of all supergradients at $\mathbf{x} \in \mathbb{R}^n$ is denoted as $\partial g(\mathbf{x})$. Since $g_1(\cdot)$ is concave on $[0, \infty)$, using the inequality (21), we have

$$\begin{aligned} g_1(\bar{\sigma}_{ij}(\mathcal{A})) &\leq g_1(\bar{\sigma}_{ij}(\mathcal{A}^{[t]})) \\ &\quad + w_{ij}^{\{t\}} (\bar{\sigma}_{ij}(\mathcal{A}) - \bar{\sigma}_{ij}(\mathcal{A}^{[t]})), \end{aligned} \quad (22)$$

where the superscript t denotes the t -th iteration results and $w_{ij}^{\{t\}} \in \partial g_1(\bar{\sigma}_{ij}(\mathcal{A}^{[t]}))$. Since $\bar{\sigma}_{1j}(\mathcal{A}^{[t]}) \geq \bar{\sigma}_{2j}(\mathcal{A}^{[t]}) \geq \dots \geq \bar{\sigma}_{n_{(2)}j}(\mathcal{A}^{[t]}) \geq 0$, $j = 1, \dots, n_3$, based on the monotonically decreasing property on $[0, \infty)$ of the supergradient of $g_1(\cdot)$, we can conclude that the condition $0 \leq w_{1j} \leq w_{2j} \leq \dots \leq w_{n_{(2)}j}$, $j = 1, \dots, n_3$ in Theorem 2 is satisfied.

Our next step is to provide the definition of the weighted tensor ℓ_1 -norm, followed by the weighted tensor ℓ_1 -norm minimization problem and its solver.

Definition 9 (Weighted tensor ℓ_1 -norm) For an entry-wise sparse tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $h(\cdot) = \mathbf{1}$, the weighted tensor ℓ_1 -norm with the ijk -th weight parameter \dot{w}_{ijk} is denoted as

$$\|\mathcal{A}\|_{\dot{w},1} = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \sum_{k=1}^{n_3} \dot{w}_{ijk} |\mathcal{A}_{ijk}|. \quad (23)$$

For a tube-wise sparse tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $h(\cdot) = \|\cdot\|_2$, the weighted tensor ℓ_1 -norm with the ij -th weight parameter \ddot{w}_{ij} is denoted as

$$\|\mathcal{A}\|_{\ddot{w},1} = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \ddot{w}_{ij} \|\mathcal{A}_{ij,:}\|_2. \quad (24)$$

When a target tensor has entry-wise sparsity and $h(\cdot) = \mathbf{1}$, the optimality of the weighted tensor ℓ_1 -norm minimization problem is established in Theorem 3.

Theorem 3 For any $\tau > 0$, $\mathcal{B} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and the weight tensor $\dot{\mathcal{W}} = (\dot{w}_{ijk})_{n_1 \times n_2 \times n_3}$, a globally optimal solution to the following problem

$$\min_{\mathcal{A}} \tau \|\mathcal{A}\|_{\dot{w},1} + \frac{1}{2} \|\mathcal{A} - \mathcal{B}\|_F^2 \quad (25)$$

is given by the weighted tensor soft thresholding (WTST)

$$\begin{aligned} \hat{\mathcal{A}} &= \Psi_{\dot{w},\tau}(\mathcal{B}) \\ &= \text{sign}(\mathcal{B}) \odot \max(0, |\mathcal{B}| - \tau \dot{\mathcal{W}}), \end{aligned} \quad (26)$$

where $\|\mathcal{A}\|_{\dot{w},1} = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \sum_{k=1}^{n_3} \dot{w}_{ijk} |\mathcal{A}_{ijk}|$ and \odot is entry-wise multiplication.

Proof To solve the ℓ_1 -minimization problem:

$$\min_a \vartheta |a| + \frac{1}{2} (b - a)^2, \quad (27)$$

Donoho [55] proposed a soft thresholding operator to obtain a globally optimal solution:

$$\hat{a} = \psi_\vartheta(b) = \text{sign}(b) \max(0, |b| - \vartheta). \quad (28)$$

Note that the defined weighted tensor ℓ_1 -norm minimization problem is obviously a variant of the ℓ_1 -minimization problem in tensor form. Since the weighted tensor ℓ_1 -norm and the tensor Frobenius norm are entry-wise norms, the WTST can be immediately obtained by letting $\vartheta = \tau \dot{w}_{ijk}$ and using the definition of the soft thresholding operator. \square

When a target tensor has structured sparsity on the tubes and $h(\cdot) = \|\cdot\|_2$, the optimality of the weighted tensor ℓ_1 -norm minimization problem is established in Corollary 1.

Algorithm 4 WTST.

Input: $\mathcal{B} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, $\dot{\mathcal{W}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ (or $\ddot{\mathcal{W}} \in \mathbb{R}^{n_1 \times n_2}$), and $\tau > 0$.
Output: $\hat{\mathcal{A}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$.

```

1: if  $h(\cdot) = \mathbf{1}$  then
2:    $\hat{\mathcal{A}} = \text{sign}(\mathcal{B}) \odot (|\mathcal{B}| - \tau \dot{\mathcal{W}})_+$ 
3: else if  $h(\cdot) = \|\cdot\|_2$  then
4:   for  $i = 1, \dots, n_1$  do
5:     for  $j = 1, \dots, n_2$  do
6:        $\hat{\mathcal{A}}_{ij,:} = \left(1 - \frac{\tau \ddot{w}_{ij}}{\|\mathcal{B}_{ij,:}\|_2}\right)_+ \cdot \mathcal{B}_{ij,:}$ 
7:     end for
8:   end for
9: end if

```

Corollary 1 For any $\tau > 0$, $\mathcal{B} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and the weight matrix $\ddot{\mathcal{W}} = (\ddot{w}_{ij})_{n_1 \times n_2}$, a globally optimal solution to the following problem

$$\min_{\mathcal{A}} \tau \|\mathcal{A}\|_{\ddot{w},1} + \frac{1}{2} \|\mathcal{A} - \mathcal{B}\|_F^2 \quad (29)$$

is given by the WTST

$$\begin{aligned} \hat{\mathcal{A}}_{ij,:} &= \Psi_{\ddot{w},\tau}(\mathcal{B}) \\ &= \begin{cases} \frac{\|\mathcal{B}_{ij,:}\|_2 - \tau \ddot{w}_{ij}}{\|\mathcal{B}_{ij,:}\|_2} \mathcal{B}_{ij,:}, & \text{if } \|\mathcal{B}_{ij,:}\|_2 \geq \tau \ddot{w}_{ij} \\ 0, & \text{otherwise} \end{cases}, \end{aligned} \quad (30)$$

where $\|\mathcal{A}\|_{\ddot{w},1} = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \ddot{w}_{ij} \|\mathcal{A}_{ij,:}\|_2$.

Remark 3 There are no restrictions on the monotonicity of the weights \dot{w}_{ijk} and \ddot{w}_{ij} in Theorem 3 and Corollary 1. Here, for convenience, we adopt the update strategy with the same weight w_{ij} , that is, $\dot{w}_{ijk}^{\{t\}}$ and $\ddot{w}_{ij}^{\{t\}}$ are set to $\dot{w}_{ijk}^{\{t\}} \in \partial g_2(|\mathcal{A}_{ijk}^{\{t\}}|)$ and $\ddot{w}_{ij}^{\{t\}} \in \partial g_2(\|\mathcal{A}_{ij,:}^{\{t\}}\|_2)$, respectively.

4 Optimization algorithm

In this section, an ADMM-based optimization scheme is proposed to compute the low-tubal-rank component \mathcal{L} and the sparse component \mathcal{E} in the model (17). The convergence analysis of the proposed algorithm is then given.

4.1 An ADMM-based optimization scheme

ADMM is a popular algorithm that solves regularized optimization problems by decomposing a large global problem into a series of smaller subproblems, each of which is then easier to handle. Here, we adopt it to solve the proposed generalized nonconvex regularized tensor RPCA (17).

For the generalized nonconvex regularized tensor RPCA (17), the corresponding augmented Lagrangian function is written as follows:

$$\Upsilon(\mathcal{L}, \mathcal{E}, \mathcal{Y}, \mu) = \tilde{g}_1(\mathcal{L}) + \tilde{g}_2(\mathcal{E}) + \langle \mathcal{Y}, \mathcal{L} + \mathcal{E} - \mathcal{D} \rangle + \frac{\mu}{2} \|\mathcal{L} + \mathcal{E} - \mathcal{D}\|_F^2, \quad (31)$$

where μ is a penalty parameter used to adjust the speed of the algorithm and \mathcal{Y} is the Lagrangian multiplier tensor introduced for the constraint condition in (17). This function is obviously difficult to solve directly. Thus, by utilizing the standard framework of ADMM, to tackle this function, we implement the following iterative scheme:

a) Computing \mathcal{L}^{t+1} : The low-tubal-rank tensor \mathcal{L} of iteration $t+1$ is updated by minimizing (31) with other variables fixed. Specifically, we solve the following subproblem:

$$\mathcal{L}^{t+1} = \arg \min_{\mathcal{L}} \tilde{g}_1(\mathcal{L}) + \frac{\mu^{t+1}}{2} \left\| \mathcal{L} + \mathcal{E}^{t+1} - \mathcal{D} + \frac{\mathcal{Y}^{t+1}}{\mu^{t+1}} \right\|_F^2. \quad (32)$$

Based on the property (22), we are able to relax the term $\tilde{g}_1(\mathcal{L})$, which yields

$$\begin{aligned} \tilde{g}_1(\mathcal{L}) &\leq \frac{1}{n_3} \sum_{i=1}^{n_2} \sum_{j=1}^{n_3} \left(g_1 \left(\bar{\sigma}_{ij}(\mathcal{L}^{t+1}) \right) \right. \\ &\quad \left. + w_{ij}^{t+1} \left(\bar{\sigma}_{ij}(\mathcal{L}) - \bar{\sigma}_{ij}(\mathcal{L}^{t+1}) \right) \right). \end{aligned} \quad (33)$$

Then, we use the right-hand sides of (33) as surrogates of $\tilde{g}_1(\mathcal{L})$ of (32) and update \mathcal{L}^{t+1} by solving

$$\begin{aligned} \mathcal{L}^{t+1} &= \arg \min_{\mathcal{L}} \frac{1}{n_3} \sum_{i=1}^{n_2} \sum_{j=1}^{n_3} \left(g_1 \left(\bar{\sigma}_{ij}(\mathcal{L}^{t+1}) \right) \right. \\ &\quad \left. + w_{ij}^{t+1} \left(\bar{\sigma}_{ij}(\mathcal{L}) - \bar{\sigma}_{ij}(\mathcal{L}^{t+1}) \right) \right) \\ &\quad + \frac{\mu^{t+1}}{2} \left\| \mathcal{L} + \mathcal{E}^{t+1} - \mathcal{D} + \frac{\mathcal{Y}^{t+1}}{\mu^{t+1}} \right\|_F^2 \\ &= \arg \min_{\mathcal{L}} \frac{1}{n_3} \sum_{i=1}^{n_2} \sum_{j=1}^{n_3} w_{ij}^{t+1} \bar{\sigma}_{ij}(\mathcal{L}) \\ &\quad + \frac{\mu^{t+1}}{2} \left\| \mathcal{L} + \mathcal{E}^{t+1} - \mathcal{D} + \frac{\mathcal{Y}^{t+1}}{\mu^{t+1}} \right\|_F^2 \end{aligned}$$

$$\begin{aligned} &= \arg \min_{\mathcal{L}} \|\mathcal{L}\|_{w^{t+1}, \otimes} \\ &\quad + \frac{\mu^{t+1}}{2} \left\| \mathcal{L} + \mathcal{E}^{t+1} - \mathcal{D} + \frac{\mathcal{Y}^{t+1}}{\mu^{t+1}} \right\|_F^2. \end{aligned} \quad (34)$$

It is obvious that the nonconvex program (34) is a standard weighted tensor nuclear norm minimization problem. From Theorem 2, we can obtain a closed-form solution to (34) by using the WTSVT. In addition, the weight w_{ij}^{t+1} is updated by $w_{ij}^{t+1} \in \partial g_1(\bar{\sigma}_{ij}(\mathcal{L}^{t+1}))$.

b) Computing \mathcal{E}^{t+1} : The sparse tensor \mathcal{E} of iteration $t+1$ is updated by minimizing (31) with other variables fixed. Specifically, we solve the following subproblem:

$$\begin{aligned} \mathcal{E}^{t+1} &= \arg \min_{\mathcal{E}} \tilde{g}_2(\mathcal{E}) \\ &\quad + \frac{\mu^{t+1}}{2} \left\| \mathcal{L}^{t+1} + \mathcal{E} - \mathcal{D} + \frac{\mathcal{Y}^{t+1}}{\mu^{t+1}} \right\|_F^2. \end{aligned} \quad (35)$$

Similarly, if the tensor \mathcal{E} has entry-wise sparsity and $h(\cdot) = \mathbf{1}$, we have

$$\begin{aligned} \tilde{g}_2(\mathcal{E}) &\leq \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \sum_{k=1}^{n_3} \left(g_2 \left(|\mathcal{E}_{ijk}^{t+1}| \right) \right. \\ &\quad \left. + \dot{w}_{ijk}^{t+1} \left(|\mathcal{E}_{ijk}| - |\mathcal{E}_{ijk}^{t+1}| \right) \right), \end{aligned} \quad (36)$$

which yields

$$\begin{aligned} \mathcal{E}^{t+1} &= \arg \min_{\mathcal{E}} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \sum_{k=1}^{n_3} \dot{w}_{ijk}^{t+1} |\mathcal{E}_{ijk}| \\ &\quad + \frac{\mu^{t+1}}{2} \left\| \mathcal{L}^{t+1} + \mathcal{E} - \mathcal{D} + \frac{\mathcal{Y}^{t+1}}{\mu^{t+1}} \right\|_F^2 \\ &= \arg \min_{\mathcal{E}} \|\mathcal{E}\|_{\dot{w}^{t+1}, 1} \\ &\quad + \frac{\mu^{t+1}}{2} \left\| \mathcal{L}^{t+1} + \mathcal{E} - \mathcal{D} + \frac{\mathcal{Y}^{t+1}}{\mu^{t+1}} \right\|_F^2. \end{aligned} \quad (37)$$

Clearly, the program (37) is a standard weighted tensor ℓ_1 -norm minimization problem. From Theorem 3, we can derive a closed-form solution to (37) by utilizing the WTST. Moreover, the weight \dot{w}_{ijk}^{t+1} is updated by $\dot{w}_{ijk}^{t+1} \in \partial g_2(|\mathcal{E}_{ijk}^{t+1}|)$.

In addition, if the tensor \mathcal{E} has structured sparsity on the tubes and $h(\cdot) = \|\cdot\|_2$, we have

$$\begin{aligned} \tilde{g}_2(\mathcal{E}) &\leq \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \left(g_2 \left(\|\mathcal{E}_{ij}^{t+1}\|_2 \right) \right. \\ &\quad \left. + \ddot{w}_{ij}^{t+1} \left(\|\mathcal{E}_{ij}\|_2 - \|\mathcal{E}_{ij}^{t+1}\|_2 \right) \right). \end{aligned} \quad (38)$$

Thus,

$$\begin{aligned} \mathcal{E}^{\{t+1\}} &= \arg \min_{\mathcal{E}} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \ddot{w}_{ij}^{\{t\}} \|\mathcal{E}_{ij}\|_2 \\ &\quad + \frac{\mu^{\{t\}}}{2} \left\| \mathcal{L}^{\{t+1\}} + \mathcal{E} - \mathcal{D} + \frac{\mathcal{Y}^{\{t\}}}{\mu^{\{t\}}} \right\|_F^2 \\ &= \arg \min_{\mathcal{E}} \|\mathcal{E}\|_{\dot{w}^{\{t\}}, 1} \\ &\quad + \frac{\mu^{\{t\}}}{2} \left\| \mathcal{L}^{\{t+1\}} + \mathcal{E} - \mathcal{D} + \frac{\mathcal{Y}^{\{t\}}}{\mu^{\{t\}}} \right\|_F^2. \end{aligned} \quad (39)$$

From Corollary 1, a closed-form solution to the above program (39) can be obtained by exploiting the WTST. In addition, the weight $\ddot{w}_{ij}^{\{t+1\}}$ is updated by $\ddot{w}_{ij}^{\{t+1\}} \in \partial g_2(\|\mathcal{E}_{ij}\|_2)$.

c) Computing $\mathcal{Y}^{\{t+1\}}$ and $\mu^{\{t+1\}}$: The Lagrangian multiplier tensor and penalty parameter are updated by

$$\mathcal{Y}^{\{t+1\}} = \mathcal{Y}^{\{t\}} + \mu^{\{t\}} (\mathcal{L}^{\{t+1\}} + \mathcal{E}^{\{t+1\}} - \mathcal{D}) \quad (40)$$

and

$$\mu^{\{t+1\}} = \min\{\nu \mu^{\{t\}}, \mu_{max}\}. \quad (41)$$

In Algorithm 5, we summarize the whole procedure for solving the proposed generalized nonconvex regularized tensor RPCA (17).

Algorithm 5 Algorithm for solving generalized nonconvex regularized tensor RPCA (17).

Input: Observation tensor $\mathcal{D} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$.

- 1: Initialize $\mathcal{L}^{\{0\}} = \mathcal{E}^{\{0\}} = \mathcal{Y}^{\{0\}} = \mathbf{0}$, $\nu = 1.1$, $\mu^{\{0\}} = 1e-4$, $\mu_{max} = 1e10$, $\epsilon = 1e-8$, and $t = 0$.
 - 2: **while** no convergence **do**
 - 3: Update $\mathcal{L}^{\{t+1\}}$ by (34).
 - 4: Update the weight:
 $w_{ij}^{\{t+1\}}$ by $w_{ij}^{\{t+1\}} \in \partial g_1(\bar{\sigma}_{ij}(\mathcal{L}^{\{t+1\}}))$.
 - 5: Update $\mathcal{E}^{\{t+1\}}$ by (37) or (39).
 - 6: Update the weights:
 $\dot{w}_{ijk}^{\{t+1\}}$ by $\dot{w}_{ijk}^{\{t+1\}} \in \partial g_2(|\mathcal{E}_{ijk}^{\{t+1\}}|)$,
 $\ddot{w}_{ij}^{\{t+1\}}$ by $\ddot{w}_{ij}^{\{t+1\}} \in \partial g_2(\|\mathcal{E}_{ij}^{\{t+1\}}\|_2)$.
 - 7: Update $\mathcal{Y}^{\{t+1\}}$ by (40).
 - 8: Update $\mu^{\{t+1\}}$ by (41).
 - 9: Check the convergence conditions:
 $\|\mathcal{L}^{\{t+1\}} - \mathcal{L}^{\{t\}}\|_\infty \leq \epsilon$,
 $\|\mathcal{E}^{\{t+1\}} - \mathcal{E}^{\{t\}}\|_\infty \leq \epsilon$,
 $\|\mathcal{D} - \mathcal{L}^{\{t+1\}} - \mathcal{E}^{\{t+1\}}\|_\infty \leq \epsilon$.
 - 10: Update $t \leftarrow t + 1$.
 - 11: **end while**
- Output:** $\hat{\mathcal{L}} = \mathcal{L}^{\{t\}}$ and $\hat{\mathcal{E}} = \mathcal{E}^{\{t\}}$.

4.2 Convergence analysis

We know that the convergence of convex two-block ADMM to the global minimum is guaranteed [56]. In the case of nonconvex objective functions, the convergence analysis is more complicated in ADMM. In this work, we provide a theoretical guarantee for the good convergence of the proposed Algorithm 5 based on the ADMM framework for minimizing an objective function with two nonconvex and possibly non-smooth regularizers. In particular, we discuss only the case of entry-wise sparse corruptions, and the proof for tube-wise sparse corruptions is similar.

Below, we first prove that the iterative sequences obtained by Algorithm 5 are bounded.

Lemma 1 *The sequences $\{\mathcal{L}^{\{t\}}, \mathcal{E}^{\{t\}}, \mathcal{Y}^{\{t\}}\}$ generated by Algorithm 5 are bounded.*

Proof From (40), it follows that

$$\begin{aligned} \|\mathcal{Y}^{\{t+1\}}\|_F &= \|\mathcal{Y}^{\{t\}} + \mu^{\{t\}} (\mathcal{L}^{\{t+1\}} + \mathcal{E}^{\{t+1\}} - \mathcal{D})\|_F \\ &= \mu^{\{t\}} \left\| \frac{1}{\mu^{\{t\}}} \mathcal{Y}^{\{t\}} + \mathcal{L}^{\{t+1\}} + \mathcal{E}^{\{t+1\}} - \mathcal{D} \right\|_F \\ &= \frac{\mu^{\{t\}}}{\sqrt{n_3}} \left\| \text{bdiag} \left(\overline{\mathcal{D} - \mathcal{Y}^{\{t\}}/\mu^{\{t\}}} - \mathcal{E}^{\{t+1\}} - \mathcal{L}^{\{t+1\}} \right) \right\|_F \\ &= \frac{\mu^{\{t\}}}{\sqrt{n_3}} \left\| \text{bdiag} \left(\bar{\mathcal{H}}^{\{t\}} \right) - \text{bdiag} \left(\bar{\mathcal{L}}^{\{t+1\}} \right) \right\|_F, \end{aligned} \quad (42)$$

where the third equality uses the Remark 1, $\mathcal{H}^{\{t\}} = \mathcal{D} - \mathcal{Y}^{\{t\}}/\mu^{\{t\}} - \mathcal{E}^{\{t+1\}}$ and it has the t-SVD: $\mathcal{U}_{\mathcal{H}}^{\{t\}} * \mathcal{S}_{\mathcal{H}}^{\{t\}} * (\mathcal{V}_{\mathcal{H}}^{\{t\}})^\top$. According to Theorem 2, we have

$$\begin{aligned} \mathcal{L}^{\{t+1\}} &= \mathcal{U}_{\mathcal{H}}^{\{t\}} * \text{iftt} \left((\bar{\mathcal{S}}_{\mathcal{H}}^{\{t\}} - \mathcal{W}^{\{t\}}/\mu^{\{t\}})_+, [\], 3 \right) * (\mathcal{V}_{\mathcal{H}}^{\{t\}})^\top, \end{aligned} \quad (43)$$

which will result in

$$\begin{aligned} \|\mathcal{Y}^{\{t+1\}}\|_F &= \frac{\mu^{\{t\}}}{\sqrt{n_3}} \left\| \text{bdiag} \left(\bar{\mathcal{U}}_{\mathcal{H}}^{\{t\}} \right) \cdot \left(\text{bdiag} \left(\bar{\mathcal{S}}_{\mathcal{H}}^{\{t\}} \right) \right. \right. \\ &\quad \left. \left. - \text{bdiag} \left((\bar{\mathcal{S}}_{\mathcal{H}}^{\{t\}} - \mathcal{W}^{\{t\}}/\mu^{\{t\}})_+ \right) \right) \cdot \text{bdiag} \left((\bar{\mathcal{V}}_{\mathcal{H}}^{\{t\}})^\top \right) \right\|_F \end{aligned} \quad (44)$$

Due to the orthogonal invariance of the Frobenius norm, we have

$$\begin{aligned} & \|\mathcal{Y}^{t+1}\|_F \\ &= \frac{\mu^{t+1}}{\sqrt{n_3}} \left\| \text{bdiag}(\bar{\mathcal{S}}_{\mathcal{H}}^{t+1}) - \text{bdiag}((\bar{\mathcal{S}}_{\mathcal{H}}^{t+1} - \mathcal{W}^{t+1}/\mu^{t+1})_+) \right\|_F \\ &\leq \frac{1}{\sqrt{n_3}} \sqrt{\sum_{i=1}^{n(2)} \sum_{j=1}^{n_3} (w_{ij}^{t+1})^2}. \end{aligned} \quad (45)$$

Thus, (45) shows that the sequence $\{\mathcal{Y}^t\}$ generated by Algorithm 5 has an upper bound.

Notice that the minimizer \mathcal{L}^{t+1} of (32) and the minimizer \mathcal{E}^{t+1} of (35) satisfy the following inequalities:

$$\begin{aligned} & \Upsilon(\mathcal{L}^{t+1}, \mathcal{E}^{t+1}, \mathcal{Y}^t, \mu^t) \\ &\leq \Upsilon(\mathcal{L}^t, \mathcal{E}^t, \mathcal{Y}^t, \mu^t). \end{aligned} \quad (46)$$

Moreover, we have

$$\begin{aligned} & \Upsilon(\mathcal{L}^t, \mathcal{E}^t, \mathcal{Y}^t, \mu^t) \\ &= \Upsilon(\mathcal{L}^t, \mathcal{E}^t, \mathcal{Y}^{t-1}, \mu^{t-1}) \\ &+ \frac{\mu^t - \mu^{t-1}}{2} \|\mathcal{L}^t + \mathcal{E}^t - \mathcal{D}\|_F \\ &+ \langle \mathcal{Y}^t - \mathcal{Y}^{t-1}, \mathcal{L}^t + \mathcal{E}^t - \mathcal{D} \rangle \\ &= \Upsilon(\mathcal{L}^t, \mathcal{E}^t, \mathcal{Y}^{t-1}, \mu^{t-1}) \\ &+ \frac{\mu^t - \mu^{t-1}}{2} \|(\mathcal{Y}^t - \mathcal{Y}^{t-1})/\mu^{t-1}\|_F \\ &+ \langle \mathcal{Y}^t - \mathcal{Y}^{t-1}, (\mathcal{Y}^t - \mathcal{Y}^{t-1})/\mu^{t-1} \rangle \\ &= \Upsilon(\mathcal{L}^t, \mathcal{E}^t, \mathcal{Y}^{t-1}, \mu^{t-1}) \\ &+ \frac{\mu^t + \mu^{t-1}}{2(\mu^{t-1})^2} \|\mathcal{Y}^t - \mathcal{Y}^{t-1}\|_F^2. \end{aligned} \quad (47)$$

Since $\{\mathcal{Y}^t\}$ is bounded, $\|\mathcal{Y}^t - \mathcal{Y}^{t-1}\|_F^2$ is also bounded. Denoting the maximum value of $\|\mathcal{Y}^t - \mathcal{Y}^{t-1}\|_F^2$ as κ for all $t = 1, \dots, \infty$, then combining (46) and (47), we obtain

$$\begin{aligned} & \Upsilon(\mathcal{L}^{t+1}, \mathcal{E}^{t+1}, \mathcal{Y}^t, \mu^t) \\ &\leq \Upsilon(\mathcal{L}^t, \mathcal{E}^t, \mathcal{Y}^0, \mu^0) + \kappa \sum_{t=1}^{\infty} \frac{\mu^t + \mu^{t-1}}{2(\mu^{t-1})^2}. \end{aligned} \quad (48)$$

From (41), it is clear that $\mu^t = v\mu^{t-1} = v^t\mu^0$. In Algorithm 5, we set $v = 1.1$, $\mu^0 = 1e-4$, then

$$\sum_{t=1}^{\infty} \frac{\mu^t + \mu^{t-1}}{2(\mu^{t-1})^2} = \frac{v+1}{2\mu^0} \sum_{t=1}^{\infty} \frac{1}{v^{t-1}} = \frac{v(v+1)}{2\mu^0(v-1)} \quad (49)$$

is bounded, and hence $\Upsilon(\mathcal{L}^{t+1}, \mathcal{E}^{t+1}, \mathcal{Y}^t, \mu^t)$ has an upper bound. Furthermore, we have

$$\begin{aligned} & \Upsilon(\mathcal{L}^{t+1}, \mathcal{E}^{t+1}, \mathcal{Y}^t, \mu^t) + \frac{1}{2\mu^t} \|\mathcal{Y}^t\|_F \\ &= \tilde{g}_1(\mathcal{L}^{t+1}) + \tilde{g}_2(\mathcal{E}^{t+1}) \\ &+ \frac{\mu^t}{2} \left\| \mathcal{L}^{t+1} + \mathcal{E}^{t+1} - \mathcal{D} + \frac{\mathcal{Y}^t}{\mu^t} \right\|_F^2. \end{aligned} \quad (50)$$

Note that all the existing nonconvex surrogates in Table 1 are monotonically increasing on $[0, \infty)$ and obey $g_1(0) = g_2(0) = 0$. Thus, each term on the right side of the above equation (50) is nonnegative. Therefore, we conclude that the sequences $\{\mathcal{L}^t\}$ and $\{\mathcal{E}^t\}$ generated by Algorithm 5 are bounded. This completes the proof. \square

The following Bolzano-Weierstrass theorem is a fundamental result of convergence in a finite-dimensional Euclidean space.

Lemma 2 (Bolzano-Weierstrass theorem) *Every bounded sequence of real numbers has a convergent subsequence.*

Now that we have the lemmas above, we can provide the convergence result for the proposed Algorithm 5.

Theorem 4 *The sequences $\{\mathcal{L}^t\}$, $\{\mathcal{E}^t\}$ generated by Algorithm 5 satisfy the following properties:*

1. $\lim_{t \rightarrow \infty} \|\mathcal{L}^{t+1} - \mathcal{L}^t\|_F = 0$.
2. $\lim_{t \rightarrow \infty} \|\mathcal{E}^{t+1} - \mathcal{E}^t\|_F = 0$.
3. $\lim_{t \rightarrow \infty} \|\mathcal{D} - \mathcal{L}^{t+1} - \mathcal{E}^{t+1}\|_F = 0$.

Proof As shown in Lemma 1, $\{\mathcal{L}^t, \mathcal{E}^t, \mathcal{Y}^t\}$ are bounded. Thus, with Lemma 2, we know that there is at least one convergent subsequence in $\{\mathcal{L}^t, \mathcal{E}^t, \mathcal{Y}^t\}$, which means that $\{\mathcal{L}^t, \mathcal{E}^t, \mathcal{Y}^t\}$ must have at least one accumulation point. By (40), we have

$$\begin{aligned} & \lim_{t \rightarrow \infty} \|\mathcal{D} - \mathcal{L}^{t+1} - \mathcal{E}^{t+1}\|_F \\ &= \lim_{t \rightarrow \infty} \|\mathcal{Y}^{t+1} - \mathcal{Y}^t\|_F / \mu^t = 0. \end{aligned} \quad (51)$$

Using (40) again and based on Theorem 2 and the t-SVD of $\mathcal{H}^{t-1}: \mathcal{U}_{\mathcal{H}}^{t-1} * \mathcal{S}_{\mathcal{H}}^{t-1} * (\mathcal{V}_{\mathcal{H}}^{t-1})^\top$, we have

$$\mathcal{L}^{t+1} = \mathcal{D} - \mathcal{E}^{t+1} + (\mathcal{Y}^{t+1} - \mathcal{Y}^t)/\mu^t \quad (52)$$

and

$$\begin{aligned} \mathcal{L}^{\{t\}} &= \mathcal{U}_{\mathcal{H}}^{\{t-1\}} \\ * \text{iff} &\left((\bar{\mathcal{S}}_{\mathcal{H}}^{\{t-1\}} - \mathcal{W}^{\{t-1\}}/\mu^{\{t-1\}})_+ [], 3 \right) * (\mathcal{V}_{\mathcal{H}}^{\{t-1\}})^{\top}. \end{aligned} \quad (53)$$

According to (52), we have

$$\begin{aligned} &\lim_{t \rightarrow \infty} \|\mathcal{L}^{\{t+1\}} - \mathcal{L}^{\{t\}}\|_F \\ &= \lim_{t \rightarrow \infty} \|\mathcal{D} - \mathcal{E}^{\{t+1\}} + (\mathcal{Y}^{\{t+1\}} - \mathcal{Y}^{\{t\}})/\mu^{\{t\}} - \mathcal{L}^{\{t\}}\|_F \\ &= \lim_{t \rightarrow \infty} \|(\mathcal{H}^{\{t\}} - \mathcal{L}^{\{t\}}) + \mathcal{Y}^{\{t+1\}}/\mu^{\{t\}}\|_F \\ &\leq \lim_{t \rightarrow \infty} \|\mathcal{H}^{\{t\}} - \mathcal{L}^{\{t\}}\|_F + \lim_{t \rightarrow \infty} \|\mathcal{Y}^{\{t+1\}}/\mu^{\{t\}}\|_F \end{aligned} \quad (54)$$

where the inequality holds because the Frobenius norm satisfies the triangular inequality. By combining (53) and the fact that $\lim_{t \rightarrow \infty} \|\mathcal{Y}^{\{t+1\}}/\mu^{\{t\}}\|_F = 0$, we can obtain

$$\begin{aligned} &\lim_{t \rightarrow \infty} \|\mathcal{L}^{\{t+1\}} - \mathcal{L}^{\{t\}}\|_F \\ &\leq \frac{1}{\sqrt{n_3}} \lim_{t \rightarrow \infty} \left\| \text{bdiag} \left(\bar{\mathcal{S}}_{\mathcal{H}}^{\{t-1\}} \right) \right. \\ &\quad \left. - \text{bdiag} \left((\bar{\mathcal{S}}_{\mathcal{H}}^{\{t-1\}} - \mathcal{W}^{\{t-1\}}/\mu^{\{t-1\}})_+ \right) \right\|_F \\ &= \frac{1}{\sqrt{n_3}} \lim_{t \rightarrow \infty} \sqrt{\sum_{i=1}^{n_{(2)}} \sum_{j=1}^{n_3} (w_{ij}^{\{t-1\}}/\mu^{\{t-1\}})^2} \\ &= 0. \end{aligned} \quad (55)$$

By Theorem 3, we know that

$$\mathcal{E}^{\{t+1\}} = \Psi_{\dot{w}^{\{t\}}, \frac{1}{\mu^{\{t\}}}}(\mathcal{M}^{\{t\}}), \quad (56)$$

where $\mathcal{M}^{\{t\}} = \mathcal{D} - \mathcal{L}^{\{t\}} - \mathcal{Y}^{\{t\}}/\mu^{\{t\}}$. Recall from (40), we have

$$\begin{aligned} &\lim_{t \rightarrow \infty} \|\mathcal{E}^{\{t+1\}} - \mathcal{E}^{\{t\}}\|_F \\ &= \lim_{t \rightarrow \infty} \|\Psi_{\dot{w}^{\{t\}}, \frac{1}{\mu^{\{t\}}}}(\mathcal{M}^{\{t\}}) - \mathcal{M}^{\{t\}} + \mathcal{L}^{\{t+1\}} - \mathcal{L}^{\{t\}} \\ &\quad - \mathcal{Y}^{\{t\}}/\mu^{\{t\}} - (\mathcal{Y}^{\{t\}} - \mathcal{Y}^{\{t-1\}})/\mu^{\{t-1\}}\|_F \\ &= \lim_{t \rightarrow \infty} \|\Psi_{\dot{w}^{\{t\}}, \frac{1}{\mu^{\{t\}}}}(\mathcal{M}^{\{t\}}) - \mathcal{M}^{\{t\}}\|_F \\ &\quad + \lim_{t \rightarrow \infty} \|\mathcal{L}^{\{t+1\}} - \mathcal{L}^{\{t\}}\|_F + \lim_{t \rightarrow \infty} \|\mathcal{Y}^{\{t\}}/\mu^{\{t\}}\|_F \\ &\quad + \lim_{t \rightarrow \infty} \|(\mathcal{Y}^{\{t\}} - \mathcal{Y}^{\{t-1\}})/\mu^{\{t-1\}}\|_F \end{aligned} \quad (57)$$

Note that

$$\begin{aligned} \lim_{t \rightarrow \infty} \|\mathcal{L}^{\{t+1\}} - \mathcal{L}^{\{t\}}\|_F &= \lim_{t \rightarrow \infty} \|\mathcal{Y}^{\{t\}}/\mu^{\{t\}}\|_F \\ &= \lim_{t \rightarrow \infty} \|(\mathcal{Y}^{\{t\}} - \mathcal{Y}^{\{t-1\}})/\mu^{\{t-1\}}\|_F = 0 \end{aligned} \quad (58)$$

Thus,

$$\begin{aligned} &\lim_{t \rightarrow \infty} \|\mathcal{E}^{\{t+1\}} - \mathcal{E}^{\{t\}}\|_F \\ &= \lim_{t \rightarrow \infty} \|\Psi_{\dot{w}^{\{t\}}, \frac{1}{\mu^{\{t\}}}}(\mathcal{M}^{\{t\}}) - \mathcal{M}^{\{t\}}\|_F \\ &= \lim_{t \rightarrow \infty} \frac{n_1 n_2 n_3 \dot{w}^{\{t\}}}{\mu^{\{t\}}} = 0, \end{aligned} \quad (59)$$

where the last equation holds because $\dot{w}^{\{t\}}$ is nonnegative and monotonically decreasing, which is attributed to g_2 being concave and monotonically increasing over $[0, \infty)$. This completes the proof. \square

5 Numerical experiments and applications

In this section, we will present some synthetic experiments and real experiments on visual inpainting to evaluate the performance of the proposed approach called GNR (the abbreviation of generalized nonconvex regularization) or GNRT (for tube-wise sparsity). For visual inpainting tasks, we compare our method with four popular competing algorithms for solving the tensor RPCA problem, which include SNN based on Tucker decomposition [57], TNN [6], PSTNN [12] and TRPCA [23] based on t-SVD. All but PSTNN are convex algorithms. TRPCA is used as a baseline for comparison only in experiments aimed at removing tube-wise sparse noise. In addition, our code will be released online at <https://github.com/zfmath>. All experiments are run in MATLAB R2016a on a 64-bit PC with an E7-4820 2.00 GHz CPU and 64 GB memory.

In the experiments, for the sake of simplicity, $g_1(\cdot)$ and $g_2(\cdot)$ in $\tilde{g}_1(\cdot)$ and $\tilde{g}_2(\cdot)$ take SCAD and ℓ_p -norm ($0 < p < 1$), respectively, as special cases of GNR and GNRT. The optimal choice of the parameters in $g_1(\cdot)$ and $g_2(\cdot)$ is made from a candidate set, and the one that consistently yields good results is used. In particular, we set $\lambda = 1$ and $\gamma = 100$ in SCAD and $\lambda = 1/\sqrt{n_{(1)} n_3}$ and $p = 0.5$ in ℓ_p . All the involved parameters in the comparison algorithms are properly calibrated by default settings to ensure the best possible performance unless otherwise specified. Some quantitative quality indices, including the relative error of \mathcal{L} (RelErrorL), the relative error of \mathcal{E} (RelErrorE), the peak signal-to-noise ratio (PSNR), the structural similarity (SSIM) and the CPU running time, are employed to evaluate the performance of

Table 3 Correct recovery using GNR for random problems

$r = \text{rank}_t(\mathcal{L}_0) = 0.05n, m_1 = \ \mathcal{E}_0\ _0 = \rho_1 n^3, \rho_1 = 0.05$						
n	r	m	$\text{rank}_t(\hat{\mathcal{L}})$	$\ \hat{\mathcal{E}}\ _0$	RelErrorL	RelErrorE
80	4	25600	4	25600	6.1e-07	2.6e-09
160	8	204800	8	204800	1.1e-07	1.9e-09
$r = \text{rank}_t(\mathcal{L}_0) = 0.05n, m_1 = \ \mathcal{E}_0\ _0 = \rho_1 n^3, \rho_1 = 0.1$						
n	r	m	$\text{rank}_t(\hat{\mathcal{L}})$	$\ \hat{\mathcal{E}}\ _0$	RelErrorL	RelErrorE
80	4	51200	4	51200	6.5e-07	2.7e-09
160	8	409600	8	409600	1.2e-06	2.0e-09
$r = \text{rank}_t(\mathcal{L}_0) = 0.1n, m_1 = \ \mathcal{E}_0\ _0 = \rho_1 n^3, \rho_1 = 0.1$						
n	r	m	$\text{rank}_t(\hat{\mathcal{L}})$	$\ \hat{\mathcal{E}}\ _0$	RelErrorL	RelErrorE
80	8	51200	8	51200	4.4e-07	2.5e-09
160	16	409600	16	409600	9.7e-07	2.4e-09

tensor recovery. RelErrorL and RelErrorE are defined as follows:

$$\text{RelErrorL} = \frac{\|\hat{\mathcal{L}} - \mathcal{L}\|_F}{\|\mathcal{L}\|_F}$$

$$\begin{aligned} \text{SSIM} \\ (60) \quad &= \frac{(2\mu_{\mathcal{C}}\mu_{\hat{\mathcal{C}}} + (0.01\varpi)^2)(2\sigma_{\mathcal{C},\hat{\mathcal{C}}} + (0.03\varpi)^2)}{(\mu_{\mathcal{C}}^2 + \mu_{\hat{\mathcal{C}}}^2 + (0.01\varpi)^2)(\sigma_{\mathcal{C}}^2 + \sigma_{\hat{\mathcal{C}}}^2 + (0.03\varpi)^2)}, \end{aligned} \quad (63)$$

and

$$\text{RelErrorE} = \frac{\|\hat{\mathcal{E}} - \mathcal{E}\|_F}{\|\mathcal{E}\|_F}. \quad (61)$$

PSNR and SSIM are defined as

$$\text{PSNR} = 10 \log_{10} \left(\frac{\|\mathcal{C}\|_\infty^2}{\frac{1}{n_1 \times n_2 \times n_3} \|\hat{\mathcal{C}} - \mathcal{C}\|_F^2} \right) \quad (62)$$

respectively, where $\mu_{\mathcal{C}}, \mu_{\hat{\mathcal{C}}}, \sigma_{\mathcal{C}}, \sigma_{\hat{\mathcal{C}}}, \sigma_{\mathcal{C},\hat{\mathcal{C}}}$ and ϖ are the local means, standard deviations, cross-covariance and dynamic range of the pixel values for tensors $\mathcal{C}, \hat{\mathcal{C}}$. Lower RelErrorL and RelErrorE and higher PSNR and SSIM values imply better recovery performance.

5.1 Synthetic experiments

5.1.1 Exact recovery for random problems

We first put the proposed method to the test on problems that are produced at random. We simply consider the ten-

Table 4 Correct recovery using GNRt for random problems

$r = \text{rank}_t(\mathcal{L}_0) = 0.05n, m_1 = \ \mathcal{E}_0\ _0 = \rho_1 n^3, m_2 = \ \mathcal{E}_{0,ij}\ _{F,0} = \rho_2 n^2, \rho_1 = \rho_2 = 0.05$						
n	r	m_1	m_2	$\text{rank}_t(\hat{\mathcal{L}})$	$\ \hat{\mathcal{E}}\ _0$	$\ \hat{\mathcal{E}}_{ij}\ _{F,0}$
80	4	25600	320	4	25600	320
160	8	204800	1280	8	204800	1280
$r = \text{rank}_t(\mathcal{L}_0) = 0.05n, m_1 = \ \mathcal{E}_0\ _0 = \rho_1 n^3, m_2 = \ \mathcal{E}_{0,ij}\ _{F,0} = \rho_2 n^2, \rho_1 = \rho_2 = 0.1$						
n	r	m_1	m_2	$\text{rank}_t(\hat{\mathcal{L}})$	$\ \hat{\mathcal{E}}\ _0$	$\ \hat{\mathcal{E}}_{ij}\ _{F,0}$
80	4	51200	640	4	51200	640
160	8	409600	2560	8	409600	2560
$r = \text{rank}_t(\mathcal{L}_0) = 0.1n, m_1 = \ \mathcal{E}_0\ _0 = \rho_1 n^3, m_2 = \ \mathcal{E}_{0,ij}\ _{F,0} = \rho_2 n^2, \rho_1 = \rho_2 = 0.1$						
n	r	m_1	m_2	$\text{rank}_t(\hat{\mathcal{L}})$	$\ \hat{\mathcal{E}}\ _0$	$\ \hat{\mathcal{E}}_{ij}\ _{F,0}$
80	8	51200	640	8	51200	640
160	16	409600	2560	16	409600	2560

Fig. 5 Fraction of correct recoveries across 10 trials for various tubal ranks r and entry-wise and tube-wise sparsity levels ρ_1 and ρ_2

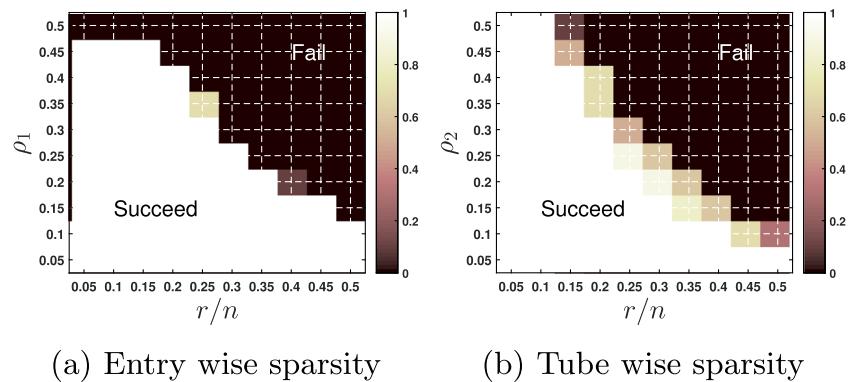


Fig. 6 20 numbered color images of size $321 \times 481 \times 3$ from the Berkeley Segmentation Dataset used in the experiments

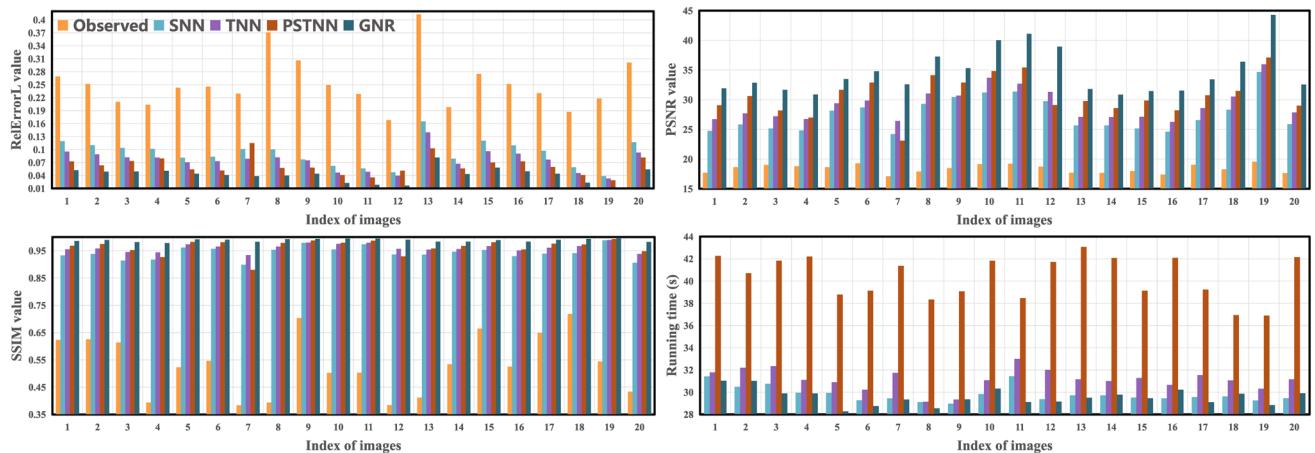


Fig. 7 Bar charts of RelErrorL, PSNR, SSIM values and running time of various tensor RPCA methods for color image inpainting with a 10% corrupted ratio

Table 5 The average values of RelErrorL, PSNR, SSIM and running time of various tensor RPCA methods for color image inpainting with 10%, 20% and 30% corrupted ratios

Method	$\rho_1 = 0.1$				$\rho_1 = 0.2$				$\rho_1 = 0.3$			
	RelErrorL	PSNR	SSIM	Time (s)	RelErrorL	PSNR	SSIM	Time (s)	RelErrorL	PSNR	SSIM	Time (s)
Observed	0.253	18.39	0.534	—	0.357	15.39	0.374	—	0.438	13.61	0.277	—
SNN [57]	0.092	27.50	0.943	29.82	0.106	26.19	0.916	28.26	0.127	24.62	0.865	27.98
TNN [6]	0.075	29.19	0.961	31.15	0.092	27.40	0.935	29.60	0.115	25.44	0.885	29.11
PSTNN [12]	0.099	30.67	0.964	40.37	0.063	27.00	0.913	41.02	0.152	23.23	0.810	41.72
GNR	0.042	34.65	0.988	29.57	0.065	30.51	0.955	29.77	0.100	26.60	0.874	30.31

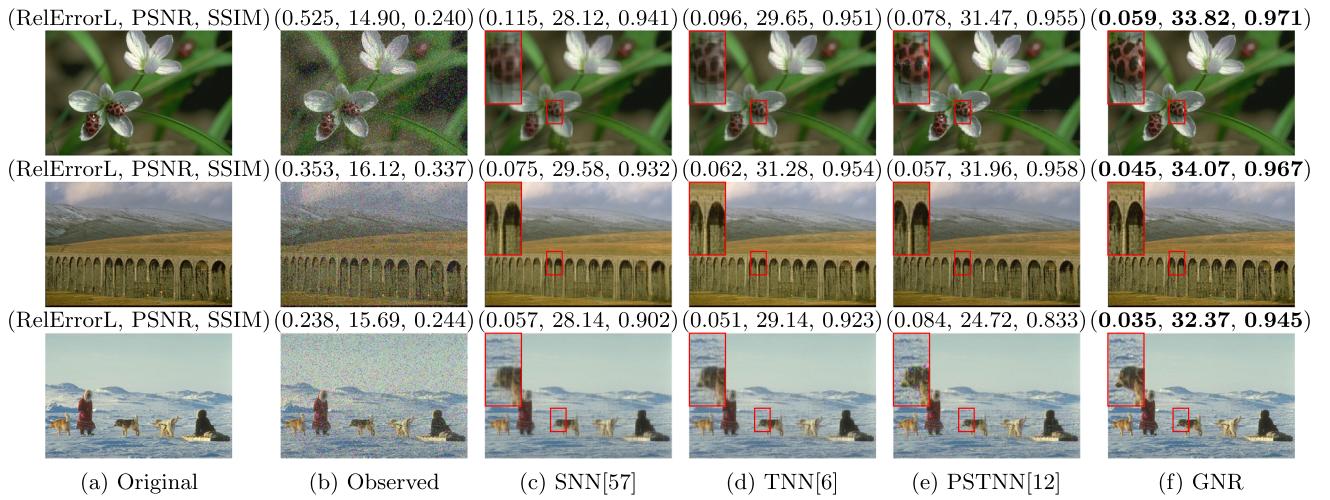


Fig. 8 Visual inpainting results of various tensor RPCA methods on color images with a 20% corrupted ratio. (a) shows the original images. (b) shows the observed images with noise corruptions. (c)-(f) show the inpainting results of SNN, TNN, PSTNN and our GNR, respectively

sors of size $n \times n \times n$ with setting $n = 80$ and 160. We use a t-product $\mathcal{L}_0 = \mathcal{K} * \mathcal{G}$ to synthesize a ground-truth low-tubal-rank tensor \mathcal{L}_0 with setting tubal rank $r = 0.05n$ and $0.1n$, where $\mathcal{K} \in \mathbb{R}^{n \times r \times n}$ and $\mathcal{G} \in \mathbb{R}^{r \times n \times n}$ are tensors with i.i.d. $\mathcal{N}(0, 1/n)$ entries. The support set of cardinality $m_1 = \|\mathcal{E}_0\|_0 = \rho_1 n^3$ with setting $\rho_1 = 0.05$ and 0.1 is uniformly chosen at random to build an entry-wise sparse noise tensor \mathcal{E}_0 , and the values in the support set are assigned ± 1 with equal probability. Tube-wise sparse noise tensors \mathcal{E}_0 are generated by selecting a support set of cardinality $m_2 = \|\mathcal{E}_{0,ij}\|_{F,0} = \rho_2 n^2$ by setting $\rho_2 = 0.05$ and 0.1 in tube units, uniformly at random and choosing values from a Gaussian distribution with mean 0 and standard deviation 1. Therefore, we let the corrupted observations $\mathcal{D} = \mathcal{L}_0 + \mathcal{E}_0$.

In Table 3 and Table 4, we show the recovery results for varying choices of the tubal rank r of \mathcal{L}_0 and the entry-wise sparsity m_1 and the tube-wise sparsity m_2 of \mathcal{E}_0 . As seen from the tables, the tubal rank estimation of $\hat{\mathcal{L}}$ and the entry-wise and tube-wise sparsity estimations of $\hat{\mathcal{E}}$ are correct in all cases. In addition, RelErrorL and RelErrorE are both very small, less than $1e-05$ and $1e-08$, respectively. These results suggest that the proposed GNR and GNRT can achieve good performance for the low-tubal-rank plus sparse tensor recovery problem.

5.1.2 Phase transition in tubal ranks and sparsity levels

We then explore the effect that the tubal rank of \mathcal{L}_0 and the corruption ratio of \mathcal{E}_0 have on the performance of our GNR and GNRT. In this experiment, we set $n = 50$ and select r/n , ρ_1 and ρ_2 as in $[0.01 : 0.05 : 0.5]$, $[0.01 : 0.05 : 0.5]$ and $[0.01 : 0.05 : 0.5]$, respectively. For each pair $(r/n, \rho_1)$ and $(r/n, \rho_2)$, we simulate 10 random instances and claim that

a recovery trial is successful when RelErrorL² is less than $1e-03$; otherwise, recovery fails. The other settings remain the same as in the previous experiment. Figure 5 provides the fraction of correct recoveries across 10 trials for each pair, in which the white region indicates perfect recovery, and the black region indicates full fail recovery. On the graph, it is possible to see that the recovery is correct throughout a large region. We can also conclude that in order to recover accurately, the larger the underlying tubal rank of \mathcal{L}_0 is, the smaller the entry-wise and tube-wise sparsity of \mathcal{E}_0 should be.

5.2 Real data experiments

5.2.1 Color image inpainting

It is well known that a color image with R, G, and B channels can be well approximated by a third-order low-rank tensor. Therefore, in this section, we consider removing random entry-wise sparse outliers from corrupted color images to evaluate the inpainting performance of the proposed GNR and comparison algorithms. To conduct the color image inpainting task, we chose 20 color images of size $321 \times 481 \times 3$ at random from the Berkeley Segmentation Dataset³ [58]. Figure 6 presents the 20 color images used for the test. In this experiment, we set the weighted parameter $\lambda = [15, 15, 1.5]$ recommended by Lu et al. [6] to obtain good performance for the SNN in most cases. For each color image, we randomly

² RelErrorL is usually much higher than that of the sparse component; thus, we only use the relative error of low-rank component recovery to measure the performance of algorithms from now on.

³ <https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/>

Table 6 The values of RelError, PSNR, SSIM by various tensor RPCA methods for the extended Yale Face Database B

ID	Subject	ρ_1	Observed	SNN [57]			TNN [6]			PSTNN [12]			GNR								
				RelError	PSNR	SSIM	RelError	PSNR	SSIM	Time (s)	RelError	PSNR	SSIM	Time (s)	RelError	PSNR	SSIM	Time (s)			
1	“yaleB01”	0.1	0.335	17.72	0.532	0.042	35.78	0.975	34.93	0.055	33.34	0.985	60.12	0.032	38.13	0.987	91.39	0.029	38.92	0.992	58.21
		0.15	0.411	15.95	0.448	0.071	31.17	0.919	33.54	0.062	32.42	0.980	64.67	0.091	29.04	0.951	97.75	0.038	36.56	0.983	61.47
		0.2	0.475	14.69	0.385	0.132	25.74	0.796	33.76	0.069	31.42	0.971	60.49	0.137	25.47	0.874	91.73	0.050	34.16	0.967	57.22
2	“yaleB05”	0.1	0.362	17.35	0.526	0.059	33.04	0.968	35.34	0.078	30.71	0.977	63.01	0.063	32.54	0.963	97.44	0.045	35.43	0.988	62.31
		0.15	0.443	15.60	0.448	0.087	29.73	0.900	33.64	0.083	30.14	0.971	63.57	0.137	25.77	0.891	98.58	0.053	34.06	0.978	62.00
		0.2	0.513	14.32	0.388	0.157	24.63	0.762	33.64	0.091	29.38	0.960	62.94	0.206	22.25	0.813	98.57	0.065	32.25	0.959	61.50
3	“yaleB20”	0.1	0.409	17.29	0.466	0.047	36.18	0.978	35.13	0.061	33.89	0.985	65.69	0.065	33.27	0.975	98.14	0.036	38.35	0.992	60.71
		0.15	0.501	15.53	0.386	0.070	32.62	0.925	33.88	0.066	33.18	0.981	65.71	0.118	28.06	0.937	98.74	0.043	36.87	0.985	61.81
		0.2	0.580	14.26	0.327	0.136	26.87	0.784	36.37	0.072	32.33	0.973	65.69	0.185	24.17	0.808	102.36	0.053	35.08	0.971	60.52

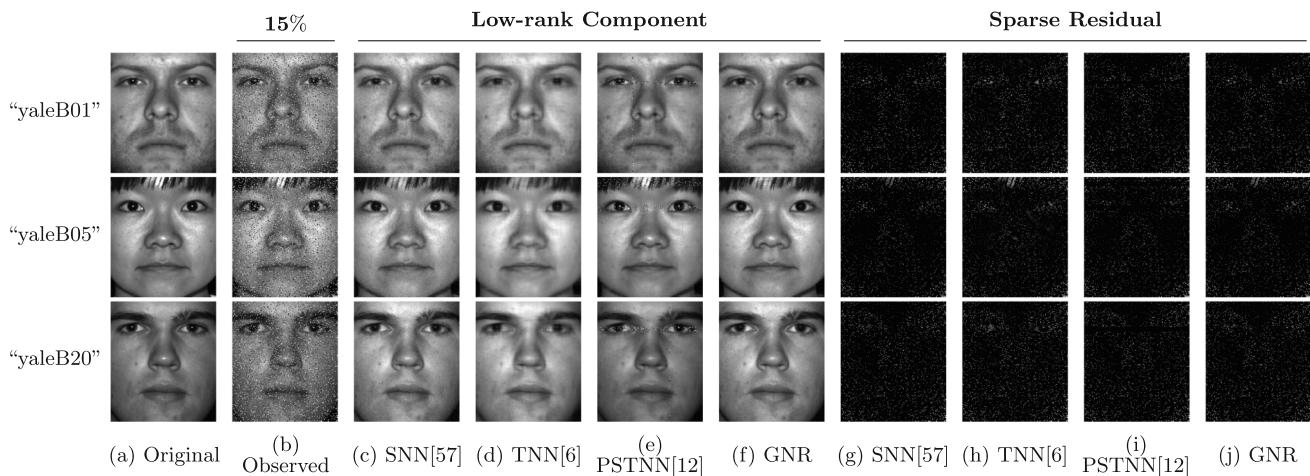


Fig. 9 Visual inpainting results of various tensor RPCA methods on “yaleB01”, “yaleB05” and “yaleB20” with a 15% corrupted ratio. (a) shows the original images. (b) shows the observed images with noise corruptions. (c)-(f) show the low-rank components recovered by the

SNN, TNN, PSTNN and GNR, respectively. (g)-(j) show the entry-wise sparse residual recovered by SNN, TNN, PSTNN and GNR, respectively

set pixels with different corrupted ratios of 10%, 20% and 30% to random values in [0, 255].

Figure 7 plots the bar charts of the four indices of RelErrorL, PSNR, SSIM and running time for each algorithm in inpainting 20 color images containing 10% corruptions. From this we can see that GNR not only achieves better index values than the competitive algorithms but also has good time efficiency. Table 5 provides the average values of RelErrorL, PSNR, SSIM and running time of various tensor RPCA methods for color image inpainting with 10%, 20% and 30% corrupted ratios. Bold text represents the best results. From an average perspective, GNR has a clear advantage in the RelErrorL, PSNR and SSIM indices, especially in the overall PSNR value, which is approximately 1.16~3.98 dB higher than the second-best approach. Even if there are some slightly inferior values, the differences between them and the best are very small. Note that the running time of our method is not always the fastest, but it is still competitive. Figure 8 shows the visual inpainting results with a 1.5x magnification of a partial area for some examples of color images with a 20% corrupted ratio. It can be seen from Fig. 8 that the proposed GNR algorithm outperforms other popular competing algorithms in terms of inpainting results and has a substantial improvement in evaluation indices.

5.2.2 Face inpainting

The extended Yale Face Database B⁴ [59, 60] contains 16128 images of 28 human subjects under 9 poses and 64

illumination conditions. In this experiment, a cropped version of the database is used to select the facial images of three subjects named “yaleB01”, “yaleB05” and “yaleB20”, each with 32 different illuminations, one fixed pose and a resolution of 192×168 . The facial image data of each subject can be constructed as a third-order tensor $192 \times 168 \times 32$. For each test tensor, we randomly set pixels with different ratios of 10%, 15% and 20% to random values in [0, 255]. Now, we evaluate the performance of our GNR and comparison algorithms on the face inpainting task. Specifically, we reshape the test facial image data to $192 \times 32 \times 168$ tensors and empirically set the weighted parameter $\lambda = [20, 0.1, 20]$ for the SNN to achieve optimal performance.

Table 6 provides the values of RelErrorL, PSNR, SSIM and running time by various tensor RPCA methods for “yaleB01”, “yaleB05” and “yaleB20” with 10%, 15% and 20% corrupted ratios. From these data, we can see that the SNN has the fastest computation rate, but the results are not satisfactory. The nonconvex PSTNN algorithm has the worst results in most cases, which may be due to the inability of PSTNN to accurately estimate the rank of this data tensor in advance. The proposed GNR algorithm received the highest evaluation value, while the time spent was almost equal to that of the TNN. In particular, in terms of overall PSNR values, our GNR algorithm improves the second-best method by approximately 0.79~4.14 dB. As shown in Fig. 9, the visualization results obtained by the proposed GNR algorithm are better than those of other algorithms in the facial image data inpainting task.

⁴ <http://vision.ucsd.edu/~iskwak/ExtYaleDatabase/ExtYaleB.html>

Table 7 The values of RelErrorL, PSNR, SSIM by various tensor RPCA methods for the YUV

ID	Subject	ρ_1	Observed	SNN [57]			TNN [6]			PSTNN [12]			GNR								
				RelError	PSNR	SSIM	RelError	PSNR	SSIM	Time (s)	RelError	PSNR	SSIM	Time (s)	RelError	PSNR					
1	“akiyo”	0.1	0.270	17.67	0.264	0.122	24.54	0.533	32.68	0.033	36.00	0.983	90.94	0.028	37.26	0.986	100.90	0.015	42.63	0.993	69.61
		0.15	0.332	15.88	0.201	0.231	19.02	0.290	32.18	0.035	35.32	0.981	90.30	0.032	36.22	0.983	101.80	0.019	40.66	0.986	73.36
		0.2	0.383	14.64	0.163	0.324	16.09	0.193	25.54	0.038	34.73	0.979	87.66	0.037	34.92	0.976	109.97	0.024	38.67	0.975	74.46
2	“coastguard”	0.1	0.218	19.21	0.360	0.122	24.23	0.592	32.92	0.047	32.48	0.949	76.89	0.041	33.66	0.953	95.73	0.030	36.55	0.971	71.81
		0.15	0.267	17.46	0.278	0.200	19.97	0.375	30.35	0.051	31.83	0.940	71.89	0.045	32.92	0.944	95.78	0.039	34.25	0.946	72.65
		0.2	0.308	16.22	0.224	0.269	17.38	0.261	26.36	0.055	31.17	0.927	73.74	0.050	32.09	0.930	95.79	0.049	32.22	0.910	71.56
3	“mobile”	0.1	0.214	18.45	0.579	0.182	19.85	0.634	25.83	0.087	26.25	0.908	78.78	0.076	27.44	0.918	94.72	0.058	29.84	0.944	74.91
		0.15	0.262	16.68	0.490	0.243	17.33	0.514	23.19	0.094	25.60	0.890	77.63	0.085	26.44	0.893	99.12	0.070	28.10	0.913	75.16
		0.2	0.303	15.43	0.423	0.293	15.71	0.432	22.03	0.102	24.89	0.866	77.62	0.098	25.19	0.855	108.60	0.085	26.44	0.871	75.42

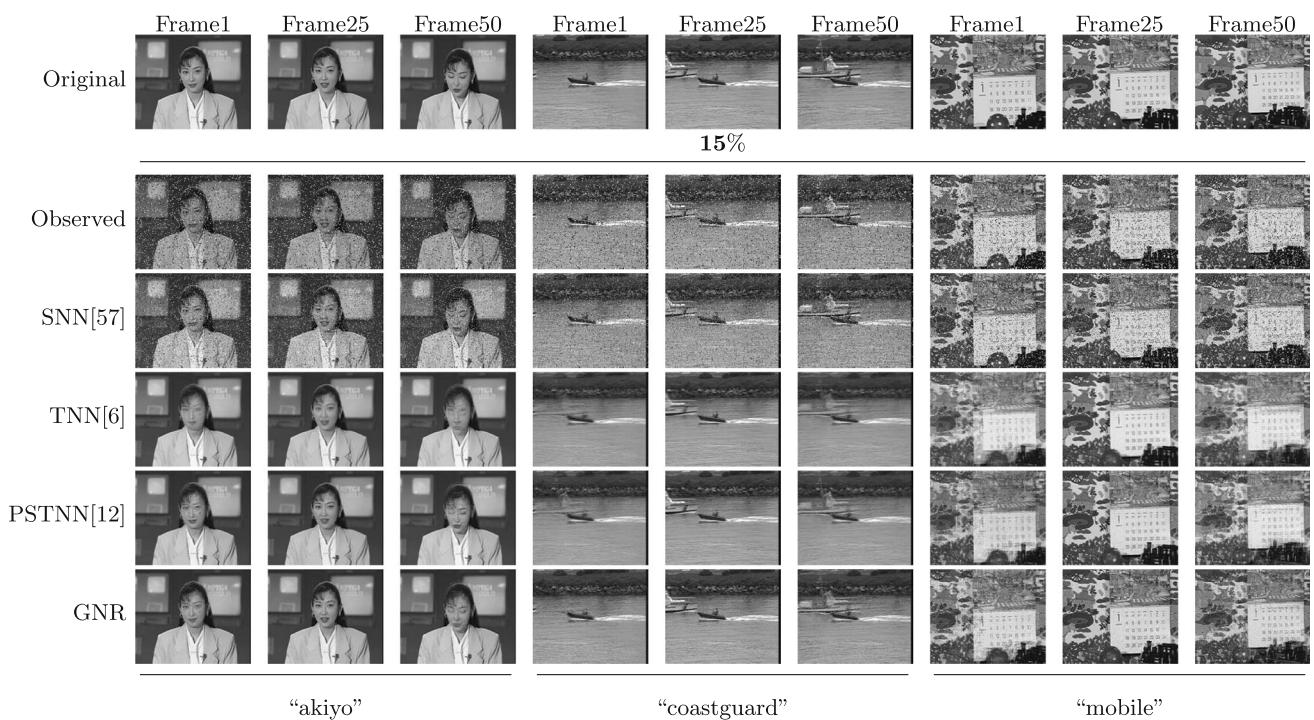


Fig. 10 Visual inpainting results of various tensor RPCA methods on “akiyo”, “coastguard” and “mobile” with a 15% corrupted ratio. The top row shows frames 1, 25 and 50 of the original video. Starting from

the second row, from top to bottom show the observed corrupted frames and the recovery frames obtained from the SNN, TNN, PSTNN, and GNR, respectively

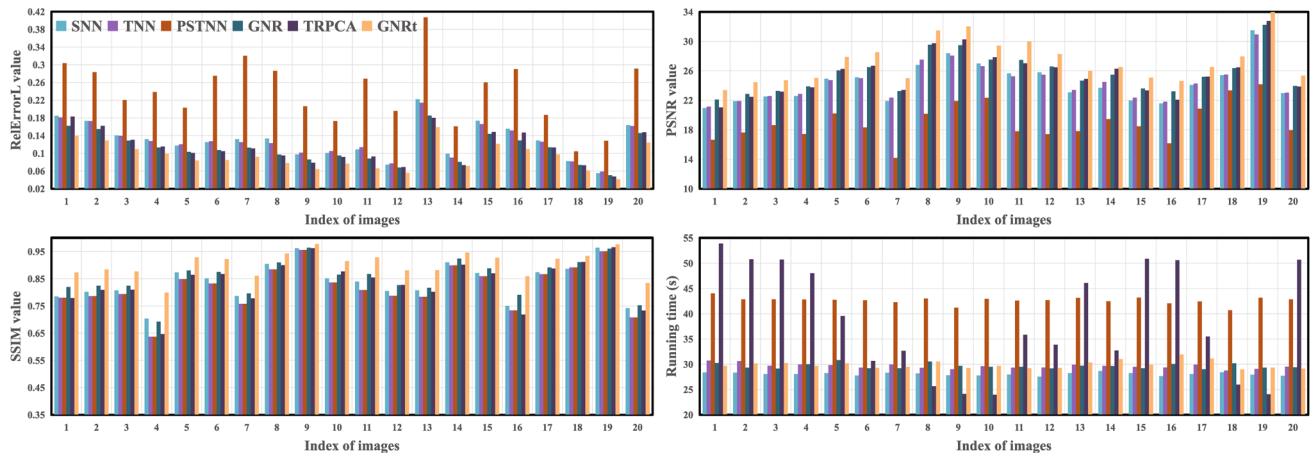


Fig. 11 Bar charts of RelErrorL, PSNR, SSIM values and running time of various tensor RPCA methods for pixel-corrupted color image inpainting with a 40% corrupted ratio

Table 8 The average values of RelErrorL, PSNR, SSIM and running time of various tensor RPCA methods for pixel-corrupted color image inpainting with 20%, 30% and 40% corrupted ratios

Method	$\rho_2 = 0.2$				$\rho_2 = 0.3$				$\rho_2 = 0.4$			
	RelErrorL	PSNR	SSIM	Time (s)	RelErrorL	PSNR	SSIM	Time (s)	RelErrorL	PSNR	SSIM	Time (s)
Observed	0.944	18.39	0.126	—	1.156	5.14	0.084	—	1.336	3.88	0.062	—
SNN [57]	0.099	26.76	0.926	28.53	0.112	25.69	0.896	28.40	0.130	24.39	0.839	28.08
TNN [6]	0.087	27.85	0.931	30.87	0.105	26.25	0.888	30.59	0.128	24.47	0.820	29.63
PSTNN [12]	0.141	23.73	0.820	42.40	0.182	21.54	0.738	42.87	0.240	19.04	0.820	42.64
GNR	0.069	29.94	0.946	39.51	0.090	27.65	0.906	29.84	0.112	25.66	0.854	29.64
TRPCA [23]	0.060	31.15	0.947	39.51	0.084	28.29	0.904	36.74	0.113	25.65	0.838	38.31
GNRt	0.056	31.83	0.964	29.44	0.074	29.32	0.937	30.07	0.093	27.31	0.904	29.92

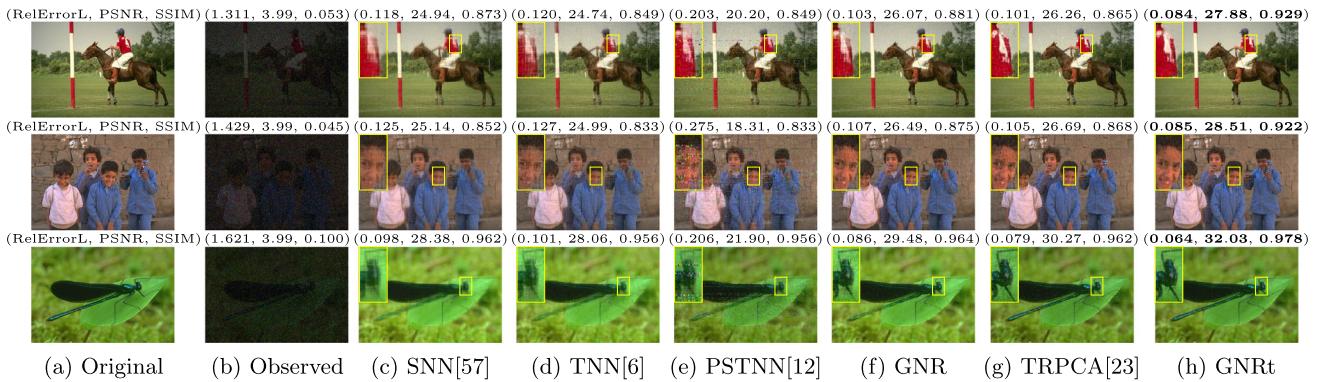


Fig. 12 Visual inpainting results of various tensor RPCA methods on pixel-corrupted color images with a 40% corrupted ratio. (a) shows the original images. (b) shows the observed images with noise corruptions. (c)-(h) show the inpainting results of SNN, TNN, PSTNN, GNR, TRPCA and our GNRT, respectively

5.2.3 Video inpainting

We use the YUV Video Sequences⁵ to further evaluate our method for the video inpainting task. Three common sequences called “akiy”, “coastguard” and “mobile” are selected for the test. Each sequence has a frame size of 144×176 , and as a result of computational limitations, only the first 50 frames of each sequence are utilized for processing. Thus, we are able to construct three test tensors of size $144 \times 176 \times 50$. We randomly set corrupted ratios of 10%, 15% and 20% for each test tensor to randomly generated values between 0 and 255.

For “akiy”, “coastguard” and “mobile” with corrupted ratios of 10%, 15% and 20%, Table 7 shows the values of RelErrorL, PSNR, SSIM, and running time by various tensor RPCA algorithms. Looking at the performance on the extended Yale Face Database B, we can see that the SNN still performs the fastest on this dataset, and the nonconvex PSTNN goes up and performs quite well. Unsurprisingly, our GNR algorithm is the strongest performer and has a major advantage over other algorithms when the corrupted ratio is 10%, especially in terms of the overall PSNR improvement of approximately $0.13\sim 5.37$ dB compared to the next best algorithm. The visualization results for frames 1, 25 and 50 of the three test videos provided in Fig. 10 further corroborate the conclusions presented in Table 7.

5.2.4 Pixel-corrupted color image inpainting

In this experiment, we still use the 20 color images shown in Fig. 6 to evaluate the proposed GNRT algorithm’s ability to remove tube-wise sparse noise. In contrast to the entry-wise corrupted color image in Section 5.2.1, we apply heavy additive noise to the pixels of the color image, which corresponds to the case where the sensor pixel point is damaged or

disturbed. We first randomly select some sparse pixel tubes along the RGB channel of the color image with different corrupted ratios of 20%, 30% and 40%, and then add random Gaussian noise to it for each selected pixel (Table 8).

A comparison of the RelErrorL, PSNR, SSIM values and running time is shown in Fig. 11 for all 20 color images with 40% corruptions. Table 8 gives the average values of four evaluation indices of various tensor RPCA methods for pixel-corrupted color image inpainting with 20%, 30% and 40% corrupted ratios. Figure 12 illustrates some examples of the recovered color images with 40% corruptions. According to these results, the following observations can be made. First, both TRPCA and GNRT outperform any other algorithm, including our proposed GNR algorithm. However, the latter exhibited even superior performance. The reason is that none of the other algorithms can effectively portray the tube-wise sparsity. Second, GRNt performs better and is less computationally expensive than TRPCA in most cases because GNRT exploits the generalized nonconvex relaxation technique. Third, when the corrupted ratio is relatively large, the advantage of GNRT is enhanced, while that of TRPCA is decreasing, such that it cannot even outperform our proposed nonconvex algorithm GNR that does not consider the tube-wise sparse structure, which indicates that our proposed GNRT is a more robust algorithm in handling the pixel-corrupted color image inpainting task.

6 Conclusion

This study set out to solve the low-tubal-rank plus sparse tensor decomposition (tensor RPCA) problem using generalized nonconvex regularization. The normal assumptions that most popular nonconvex functions can satisfy are imposed on the proposed generalized nonconvex surrogates, allowing it to greatly improve the low-tubal-rankness and sparsity of the tensor. The second aspect of this study is the incorporation of

⁵ <http://trace.eas.asu.edu/yuv/>

tube-wise sparse corruptions into the framework of generalized nonconvex optimization established above. We present a detailed procedure with an ADMM structure for solving such nonconvex problems and provide theoretical guarantees for the convergence of the method. Our extensive visual inpainting experiments demonstrate that the proposed strategy outperforms the existing methods in terms of its ability to reconstruct tensor data.

An issue that is not addressed in this study is whether the structural features of the tensor singular value vector are considered to further enhance the effectiveness of the generalized nonconvex regularization method in dealing with the tensor RPCA. One additional concern is that we have not yet provided provable accuracy guarantees for the proposed nonconvex model. Furthermore, in addition to the model-driven noise processing scheme presented in this paper, data-driven deep learning methods have made significant progress in processing noisy data due to their deep representation capabilities. These methods have demonstrated excellent results in various fields, including video processing [61], earthquake signal detection [62] and medical image analysis [63]. Integrating the proposed generalized nonconvex method with deep learning may lead to more flexible and efficient processing of practical noise problems. We raise these issues as valuable topics for further consideration in the future.

Acknowledgements This work was supported by the National Natural Science Foundation of China (Grant No. 12101512), the China Postdoctoral Science Foundation (Grant No. 2021M692681), the Natural Science Foundation of Chongqing, China (Grant No. cstc2021jcyj-bshX0155) and the Fundamental Research Funds for the Central Universities (Grant No. SWU120078).

Availability of data and materials The datasets generated and/or analyzed during the current study are available from the corresponding author on reasonable request.

References

- Wang Y, Lin L, Zhao Q, Yue T, Meng D, Leung Y (2017) Compressive sensing of hyperspectral images via joint tensor tucker decomposition and weighted total variation regularization. *IEEE Geosci Remote Sens Lett* 14(12):2457–2461
- Zheng Y, Huang T, Zhao X, Jiang T, Ma T, Ji T (2020) Mixed noise removal in hyperspectral image via low-fibered-rank regularization. *IEEE Trans Geosci Remote Sens* 58(1):734–749
- Tao D, Guo Y, Li Y, Gao X (2018) Tensor rank preserving discriminant analysis for facial recognition. *IEEE Trans Image Process* 27(1):325–334
- Frolov E, Oseledets I (2017) Tensor methods and recommender systems. *Wiley Interdiscip Rev Data Min Knowl Disc* 7(3):e1201
- Zhang Y, Bi X, Tang N, Qu A (2021) Dynamic tensor recommender systems. *J Mach Learn Res* 22(65):1–35
- Lu C, Feng J, Chen Y, Liu W, Lin Z, Yan S (2020) Tensor robust principal component analysis with a new tensor nuclear norm. *IEEE Trans Pattern Anal Mach Intell* 42(4):925–938
- Huang H, Liu Y, Long Z, Zhu C (2020) Robust low-rank tensor ring completion. *IEEE Trans Comput Imaging* 6:1117–1126
- Li B, Zhao X, Wang J, Chen Y, Jiang T, Liu J (2021) Tensor completion via collaborative sparse and low-rank transforms. *IEEE Trans Comput Imaging* 7:1289–1303
- Kilmer M, Horesh L, Avron H, Newman E (2021) Tensor-tensor algebra for optimal representation and compression of multiway data. *Proc Natl Acad Sci* 118(28)
- Zhou P, Feng J (2017) Outlier-robust tensor pca. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp 2263–2271
- Chen Y, Wang S, Zhou Y (2018) Tensor nuclear norm-based low-rank approximation with total variation regularization. *IEEE J Sel Top Signal Process* 12(6):1364–1377
- Jiang T, Huang T, Zhao X, Deng L (2020) Multi-dimensional imaging data recovery via minimizing the partial sum of tubal nuclear norm. *J Comput Appl Math* 372:112680
- Zhang F, Wang J, Wang W, Xu C (2021) Low-tubal-rank plus sparse tensor recovery with prior subspace information. *IEEE Trans Pattern Anal Mach Intell* 43(10):3492–3507
- Zhang X, Wang D, Zhou Z, Ma Y (2021) Robust low-rank tensor recovery with rectification and alignment. *IEEE Trans Pattern Anal Mach Intell* 43(1):238–255
- Xie Q, Zhao Q, Meng D, Xu Z (2018) Kronecker-basis-representation based tensor sparsity and its applications to tensor recovery. *IEEE Trans Pattern Anal Machine Intell* 40(8):1888–1902
- Chen Y, Xiao X, Peng C, Lu G, Zhou Y (2022) Low-rank tensor graph learning for multi-view subspace clustering. *IEEE Trans Circ Syst Video Technol* 32(1):92–104
- Li Z, Wang Y, Zhao Q, Zhang S, Meng D (2022) A tensor-based online rpca model for compressive background subtraction. *IEEE Trans Neural Netw Learn Syst*
- Kiers H (2000) Towards a standardized notation and terminology in multiway analysis. *J Chemom* 14(3):105–122
- Tucker L (1966) Some mathematical notes on three-mode factor analysis. *Psychometrika* 31(3):279–311
- Oseledets I (2011) Tensor-train decomposition. *SIAM. J Sci Comput* 33(5):2295–2317
- Kilmer M, Martin C (2011) Factorization strategies for third-order tensors. *Linear Algebra Appl* 435(3):641–658
- Liu J, Musialski P, Wonka P, Ye J (2013) Tensor completion for estimating missing values in visual data. *IEEE Trans Pattern Anal Mach Intell* 35(1):208–220
- Zhang Z, Ely G, Aeron S, Hao N, Kilmer M (2014) Novel methods for multilinear data completion and de-noising based on tensor-svd. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp 3842–3849
- Lu C, Feng J, Chen Y, Liu W, Lin Z, Yan S (2016) Tensor robust principal component analysis: Exact recovery of corrupted low-rank tensors via convex optimization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp 5249–5257
- Bengua J, Phien H, Tuan H, Do M (2017) Efficient tensor completion for color image and video recovery: Low-rank tensor train. *IEEE Trans Image Process* 26(5):2466–2479
- Friedland S, Lim LH (2018) Nuclear norm of higher-order tensors. *Math Comput* 87(311):1255–1281
- Yang J, Zhao X, Ji T, Ma T, Huang T (2020) Low-rank tensor train for tensor robust principal component analysis. *Appl Math Comput* 367:124783
- Candès E, Wakin M, Boyd S (2008) Enhancing sparsity by reweighted ℓ_1 minimization. *J Fourier Anal Appl* 14(5):877–905
- Gao C, Wang N, Yu Q, Zhang Z (2011) A feasible nonconvex relaxation approach to feature selection. In: Proceedings of the AAAI Conference on Artificial Intelligence. pp 356–361
- Geman D, Yang C (1995) Nonlinear image recovery with half-quadratic regularization. *IEEE Trans Image Process* 4(7):932–946

31. Trzasko J, Manduca A (2008) Highly undersampled magnetic resonance image reconstruction via homotopic ℓ_0 -minimization. *IEEE Trans Med Imaging* 28(1):106–121
32. Friedman J (2012) Fast sparse regression and classification. *Int J Forecast* 28(3):722–738
33. Frank L, Friedman J (1993) A statistical view of some chemometrics regression tools. *Technometrics* 35(2):109–135
34. Zhang C (2010) Nearly unbiased variable selection under minimax concave penalty. *Ann Stat* 38(2):894–942
35. Zhang T (2010) Analysis of multi-stage convex relaxation for sparse regularization. *J Mach Learn Res* 11(3):1081–1107
36. Fan J, Li R (2001) Variable selection via nonconcave penalized likelihood and its oracle properties. *J Am Stat Assoc* 96(456):1348–1360
37. Xue J, Zhao Y, Liao W, Chan J (2019) Nonconvex tensor rank minimization and its applications to tensor recovery. *Inf Sci* 503:109–128
38. Cai S, Luo Q, Yang M, Li W, Xiao M (2019) Tensor robust principal component analysis via non-convex low rank approximation. *Appl Sci* 9(7):1411
39. Chen Y, Wang S, Peng C, Hua Z, Zhou Y (2021) Generalized nonconvex low-rank tensor approximation for multi-view subspace clustering. *IEEE Trans Image Process* 30:4022–4035
40. Qiu D, Bai M, Ng M, Zhang X (2021) Nonlocal robust tensor recovery with nonconvex regularization. *Inverse Probl* 37(3):035001
41. Yang M, Luo Q, Li W, Xiao M (2022) Nonconvex 3d array image data recovery and pattern recognition under tensor framework. *Pattern Recogn* 122:108311
42. Lu C, Tang J, Yan S, Lin Z (2015) Nonconvex nonsmooth low rank minimization via iteratively reweighted nuclear norm. *IEEE Trans Image Process* 25(2):829–839
43. Boyd S, Parikh N, Chu E, Peleato B, Eckstein J (2011) Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found Trends Mach Learn* 3(1):1–122
44. Wang J, Hou J, Eldar Y (2022) Tensor robust principal component analysis from multilevel quantized observations. *IEEE Trans Inf Theory* 69(1):383–406
45. Wang A, Jin Z, Tang G (2020) Robust tensor decomposition via t-svd: Near-optimal statistical guarantee and scalable algorithms. *Signal Process* 167:107319
46. Lu C, Tang J, Yan S, Lin Z (2014) Generalized nonconvex nonsmooth low-rank minimization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp 4130–4137
47. Lu C, Zhu C, Xu C, Yan S, Lin Z (2015) Generalized singular value thresholding. In: Proceedings of the AAAI Conference on Artificial Intelligence. pp 1805–1811
48. Zhang X (2018) A nonconvex relaxation approach to low-rank tensor completion. *IEEE Trans Neural Netw Learn Syst* 30(6):1659–1671
49. Wen F, Ying R, Liu P, Truong T (2019) Nonconvex regularized robust pca using the proximal block coordinate descent algorithm. *IEEE Trans Signal Process* 67(20):5402–5416
50. Yang Z, Fan L, Yang Y, Yang Z, Gui G (2020) Generalized nuclear norm and laplacian scale mixture based low-rank and sparse decomposition for video foreground-background separation. *Signal Process* 172:107527
51. Liu X, Lu J, Shen L, Xu C, Xu Y (2020) Multiplicative noise removal: nonlocal low-rank model and its proximal alternating reweighted minimization algorithm. *SIAM J Imaging Sci* 13(3):1595–1629
52. Wang H, Zhang F, Wang J, Huang T, Huang J, Liu X (2021) Generalized nonconvex approach for low-tubal-rank tensor recovery. *IEEE Trans Neural Netw Learn Syst* 33(8):3305–3319
53. Zhang X, Zheng J, Zhao L, Zhou Z, Lin Z (2022) Tensor recovery with weighted tensor average rank. *IEEE Trans Neural Netw Learn Syst*
54. Kilmer M, Braman K, Hao N, Hoover R (2013) Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging. *SIAM J Matrix Anal Appl* 34(1):148–172
55. Donoho D (1995) De-noising by soft-thresholding. *IEEE Trans Inf Theory* 41(3):613–627
56. Eckstein J, Bertsekas D (1992) On the douglas-rachford splitting method and the proximal point algorithm for maximal monotone operators. *Math Program* 55(3):293–318
57. Huang B, Mu C, Goldfarb D, Wright J (2015) Provable models for robust low-rank tensor completion. *Pac J Optim* 11(2):339–364
58. Martin D, Fowlkes C, Tal D, Malik J (2001) A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proceedings Eighth IEEE International Conference on Computer Vision. pp 416–423
59. Georghiades A, Belhumeur P, Kriegman D (2001) From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans Pattern Anal Mach Intell* 23(6):643–660
60. Lee K, Ho J, Kriegman D (2005) Acquiring linear subspaces for face recognition under variable lighting. *IEEE Trans Pattern Anal Mach Intell* 27(5):684–698
61. Wielgosz M, Pietroń M (2017) Using spatial pooler of hierarchical temporal memory to classify noisy videos with predefined complexity. *Neurocomputing* 240:84–97
62. Mousavi S, Ellsworth W, Zhu W, Chuang L, Beroza G (2020) Earthquake transformer—an attentive deep-learning model for simultaneous earthquake detection and phase picking. *Nat Commun* 11(1):3952
63. Karimi D, Dou H, Warfield S, Gholipour A (2020) Deep learning with noisy labels: Exploring techniques and remedies in medical image analysis. *Med Image Anal* 65:101759

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.