Signal Processing 167 (2020) 107319

Contents lists available at ScienceDirect

Signal Processing

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

Robust tensor decomposition via t-SVD: Near-optimal statistical guarantee and scalable algorithms

Andong Wang[a,b], Zhong Jin[a,b,*], Guoqing Tang[c]

[a] School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China
[b] Key Laboratory of Intelligent Perception and System for High-Dimensional Information of Ministry of Education, Nanjing University of Science and Technology, Nanjing 210094, China
[c] Department of Mathematics, North Carolina Agricultural and Technical, State University, Greensboro, NC 27411, USA

ARTICLE INFO

Article history:
Received 30 April 2019
Revised 31 July 2019
Accepted 24 September 2019
Available online 25 September 2019

Keywords:
Tensor recovery
Tensor SVD
Low-rank recovery
Estimation error
ADMM

ABSTRACT

Aiming at recovering a signal tensor from its mixture with outliers and noises, robust tensor decomposition (RTD) arises frequently in many real-world applications. Recently, the low-tubal-rank model has shown more powerful performances than traditional tensor low-rank models in several tensor recovery tasks. Assuming the underlying tensor to be low-tubal-rank and the outliers sparse, this paper first proposes a penalized least squares estimator for RTD. Specifically, we adopt the tubal nuclear norm (TNN) and a sparsity inducing norm to regularize the underlying tensor and the outliers, respectively. Then, from a statistical standpoint, non-asymptotic upper bounds on the estimation error are established and proved to be near-optimal in a minimax sense. Further, two algorithms, namely, an ADMM-based algorithm and a Frank-Wolfe (FW) based algorithm are proposed to efficiently solve the proposed estimator from a computational standpoint. The sharpness of the proposed upper bound is verified on synthetic datasets. The superiority and efficiency of the proposed algorithms is demonstrated in experiments on real datasets.

© 2019 Elsevier B.V. All rights reserved.

1. Introduction

Tensor decomposition has become a paradigm for modern multi-way array processing [1]. Traditional tensor decomposition models like CANDECOMP/PARAFAC (CP) decomposition [2] and Tucker decomposition [3] work well when the multi-way data is mildly corrupted by small noises. However, in many applications, the multi-way data may often be corrupted by both small noises and gross outliers, due to various reasons like occlusion in videos, sensor failures, abnormalities, or software malfunctions. For example, in hyper-spectral image processing, the embedded noise is probably a mixture of small dense noise and sparse gross outliers [4]. Thus, it is of significantly practical and theoretical importance to develop efficient algorithms with performance guarantee to robustify traditional tensor decompositions.

Aiming at recovering a tensor from measurements corrupted by noises and outliers, robust tensor decomposition (RTD) [5] assumes that we observe a corrupted tensor

$$\mathcal{Y} = \mathcal{L}^* + \mathcal{S}^* + \mathcal{E}, \tag{1}$$

where $\mathcal{L}^*$ is the true but unknown signal tensor, tensor $\mathcal{S}^*$ represents outliers, and $\mathcal{E}$ denotes a (deterministic or random) noise tensor (see Fig. 1 for illustration). Here, we suppose the outlier tensor $\mathcal{S}^*$ is sparse, since it is unable to reconstruct a signal tensor when most of the measurements are heavily corrupted. In many multi-way signal processing applications like image/video processing, most studied outliers can be categorized into three possible classes, i.e., element-wise, tube-wise and slice-wise outliers, as shown in Fig. 2. The element-wise outliers are the most common in multi-media signal processing such as video restoration [6] and video surveillance [7]. The tube-wise outliers may occur when pixels of a color image are corrupted [6], and the sample-specific outliers can be modeled as slice-wisely sparse [8].

In many real-world applications, most variations of the multi-way signal can be linearly dominated by a relatively small number of latent factors due to intrinsic correlations and redundancy [9]. Such data can be well approximated by a "low rank" tensor. Since the CP rank and its corresponding nuclear norm are both NP hard [10,11], the computational efficient Tucker rank is commonly used to model real multi-way data. To recover a low-rank signal tensor

* Corresponding author at: School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China.
E-mail address: zhongjin@njust.edu.cn (Z. Jin).

https://doi.org/10.1016/j.sigpro.2019.107319
0165-1684/© 2019 Elsevier B.V. All rights reserved.

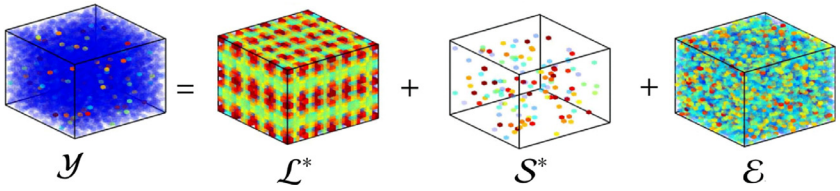


Fig. 1. Observation model of robust tensor decomposition when the outliers are element-wisely sparse.

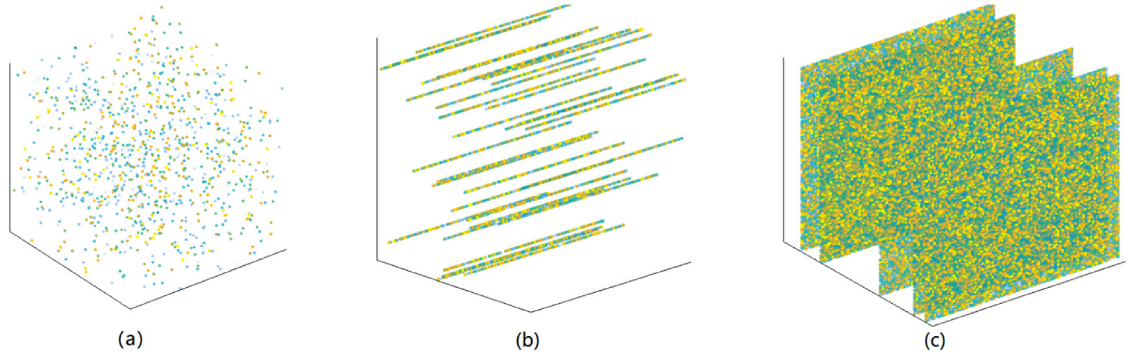


Fig. 2. Three settings of $\mathcal{S}^*$. Subplot (a): $\mathcal{S}^*$ is element-wisely sparse; Subplot (b): $\mathcal{S}^*$ is tube-wisely sparse; Subplot (c): $\mathcal{S}^*$ is slice-wisely sparse.

from noises and element-wise sparse outliers, the following RTD model is considered [5]

$$\min_{\mathcal{L},\mathcal{S}} \frac{1}{2}\|\mathcal{Y}-\mathcal{L}-\mathcal{S}\|_{\mathrm{F}}^2+\lambda\|\mathcal{L}\|_{S_1}+\mu\|\mathcal{S}\|_{l_1}, \tag{2}$$

where $\|\cdot\|_{S_1}$ is the tensor Schatten-1 norm [12] to impose low Tucker rank structure, $\|\cdot\|_{l_1}$ is the element-wise l_1-norm, and λ and μ are regularization parameters. An ADMM-based algorithm is proposed to solve the model, and the non-asymptotic estimation error of $\mathcal{L}^*$ and $\mathcal{S}^*$ are established [5].

Recently, the low tubal rank models have achieved better performances than low Tucker rank models in many low rank tensor recovery tasks, like tensor completion [13–16], tensor sensing [17,18], tensor robust principal component analysis (TRPCA) [6,19], outlier robust tensor principal component analysis (OR-TPCA) [8], etc. At the core of these models is the tubal nuclear norm (TNN) $\|\cdot\|_\star$ [20], which is pointed out to be powerful in capturing the ubiquitous "spatial-shifting" correlations in real-world multi-way data [21].

Thanks to the superioity of TNN, many relevant models are studied to recover a low rank signal tensor $\mathcal{L}^*\in\mathbb{R}^{d_1\times d_2\times d_3}$ from observation $\mathcal{Y}\in\mathbb{R}^{d_1\times d_2\times d_3}$ corrupted by sparse outliers $\mathcal{S}^*\in\mathbb{R}^{d_1\times d_2\times d_3}$ in noiseless settings (i.e., $\mathcal{E}=\mathbf{0}$ in Eq. (1)). In [6,19], a TNN-based TRPCA model is proposed for robust tensor recovery against element-wisely sparse outliers. It is proved that by solving the following problem

$$\min_{\mathcal{L},\mathcal{S}} \|\mathcal{L}\|_\star+\lambda\|\mathcal{S}^*\|_{l_1} \quad \text{s.t.} \quad \mathcal{Y}=\mathcal{L}+\mathcal{S}, \tag{3}$$

where $\lambda=1/\sqrt{\min\{d_1,d_2\}d_3}$, the true tensor $\mathcal{L}^*$ and the element-wisely sparse outlier tensor $\mathcal{S}^*$ can be exactly recovered with high probability, given $\mathcal{L}^*$ satisfies the tensor incoherence conditions. When the outliers $\mathcal{S}^*$ are tube-wisely sparse, Zhang et al. [22] proposes the following TRPCA model

$$\min_{\mathcal{L},\mathcal{S}} \|\mathcal{L}\|_\star+\lambda\|\mathcal{S}^*\|_{\mathrm{tube}_1} \quad \text{s.t.} \quad \mathcal{Y}=\mathcal{L}+\mathcal{S}, \tag{4}$$

where $\|\cdot\|_{\mathrm{tube}_1}$ is the tensor tube-1 norm (see the definition in Table 1). In [8], a slice-wisely sparse tensor $\mathcal{S}^*$ is used to represent the sample-specific outliers, and the outlier robust TPCA is proposed as follows

$$\min_{\mathcal{L},\mathcal{S}} \|\mathcal{L}\|_\star+\lambda\|\mathcal{S}^*\|_{\mathrm{slice}_1} \quad \text{s.t.} \quad \mathcal{Y}=\mathcal{L}+\mathcal{S}, \tag{5}$$

where $\|\cdot\|_{\mathrm{slice}_1}$ is the tensor slice-1 norm (see Table 1 for definition). It is proved that when $\lambda=1/\sqrt{\log d_2}$, the solution of Problem (5) can exactly recover the true tensor $\mathcal{L}^*$ and the slice-wisely sparse outliers $\mathcal{S}^*$ with high probability if $\mathcal{L}^*$ and $\mathcal{S}^*$ satisfy the tensor incoherence condition and unambiguity condition, respectively.

It is noted that Models (3)–(5) only consider the noiseless settings, i.e., $\mathcal{E}=\mathbf{0}$ in Problem (1). However, in real applications outliers and noises are more likely to coexist. On the other hand, the theoretical analysis of TRPCA [6,19] and OR-TPCA [8] assumes the underlying tensor $\mathcal{L}^*$ to satisfy the tensor incoherence conditions defined through the tensor singular value decomposition (t-SVD). Since the true $\mathcal{L}^*$ is unknown it is usually hard to check whether incoherence conditions hold. Moreover, the ADMM-based algorithms designed to solve Models (3)–(5) in [8,19,22] are computationally expensive, since they need to compute the proximity operator of TNN (which requires the time-consuming full SVDs) in each iteration.

To address the above mentioned issues, we propose a penalized least square estimator to estimate the underlying tensor $\mathcal{L}^*$ and the outlier $\mathcal{S}^*$. The theoretic analysis of this estimator does not assume the underlying tensor to satisfy the tensor incoherence conditions. Specifically, the contributions of this paper are listed as follows:

- A TNN-based least square estimator is proposed for RTD in Eq. (13). We only assume $\mathcal{L}^*$ to satisfy the l_∞-norm boundedness condition, which is less strict than the tensor incoherence conditions.
- On the statistical side, both deterministic and non-asymptotic upper bounds on the estimation error are established in Theorems 1 and 2, respectively. The non-asymptotic upper bounds are then proved to be minimax near-optimal by Theorem 3. Experiments on synthetic dataset verify that the proposed upper bounds can predict the scaling behavior of the estimation error.
- On the computational side, two algorithms, i.e., an ADMM-based algorithm (Algorithm 1) and an FW-based algorithm (Algorithm 2), are proposed with convergence guarantees (Theorems 4 and 5). The latter gets rid of the proximity operator of TNN, and has significantly cheaper one-iteration

Table 1
List of notations.

Notations	Descriptions	Notations	Descriptions
t	A scaler	$\mathcal{L}^*$	True low-rank tensor
$\mathbf{t}$	A vector	$\mathcal{S}^*$	True "sparse" tensor
$\mathbf{T}$	A matrix	$\hat{\mathcal{L}}$	Estimator of $\mathcal{L}^*$
$\mathcal{T}$	A tensor	$\hat{\mathcal{S}}$	Estimator of $\mathcal{S}^*$
$\widetilde{\mathcal{T}}$	$\text{fft}_3(\mathcal{T})$	$\|\mathcal{T}\|_{\text{sp}} := \|\overline{\mathbf{T}}\|$	Tensor spectral norm
$\overline{\mathbf{T}}$	Block-diagonal matrix of $\widetilde{\mathcal{T}}$	$\|\mathcal{T}\|_* := \|\overline{\mathbf{T}}\|_*$	Tubal nuclear norm
$\mathcal{T}_{ijk}$	$(i,j,k)_{th}$ entry of $\mathcal{T}$	$\|\mathcal{T}\|_{l_1} := \sum_{ijk}\|\mathcal{T}_{ijk}\|$	Tensor l_1-norm
$\mathcal{T}(i,j,k)$	$\mathcal{T}_{ijk}$	$\|\mathcal{T}\|_{\text{F}} := \sqrt{\sum_{ijk}\mathcal{T}_{ijk}^2}$	Tensor F-norm
$\mathcal{T}(i,j,:)$	$(i,j)_{th}$ tube of $\mathcal{T}$	$\|\mathcal{T}\|_{l_\infty} := \max_{ijk}\|\mathcal{T}_{ijk}\|$	Tensor l_∞-norm
$\mathcal{T}(:,j,:)$	j_{th} lateral slice of $\mathcal{T}$	$\|\mathcal{T}\|_{\text{tube}_1} := \sum_{ij}\|\mathcal{T}(i,j,:)\|_{\text{F}}$	Tensor tube_1-norm
$\mathcal{T}(:,:,k)$	k_{th} frontal slice of $\mathcal{T}$	$\|\mathcal{T}\|_{\text{slice}_1} := \sum_j\|\mathcal{T}(:,j,:)\|_{\text{F}}$	Tensor slice_1-norm
$\mathbf{T}^{(k)}$	$\mathcal{T}(:,:,k)$	$\|\mathcal{T}\|_{\text{tube}_\infty} := \max_{ij}\|\mathcal{T}(i,j,:)\|_{\text{F}}$	Tensor tube_∞-norm
Θ_s	Support of $\mathcal{S}^*$	$\|\mathcal{T}\|_{\text{slice}_\infty} := \max_j\|\mathcal{T}(:,j,:)\|_{\text{F}}$	Tensor slice_∞-norm
$\Theta_s^\perp$	Complement of Θ_s	$\langle\mathcal{A},\mathcal{B}\rangle := \sum_{ijk}\mathcal{A}_{ijk}\mathcal{B}_{ijk}$	Tensor inner product

computational cost. Experiments on real dataset validate the effectiveness and the efficiency of the proposed algorithms.

The rest of this paper proceeds as follows. Section 2 introduces the notations and preliminaries on tensor SVD. The proposed estimator is formulated in Section 3. We analyze the statistical performance of the proposed estimator in Section 4. Two algorithms are developed to solve the estimator in Section 5. We show experiments on both synthetic and real dataset in Section 6. Section 7 summarizes this work. Some technical proofs are given in the supplemental material.

2. Notations and preliminaries

2.1. Notations

For convenience, we list the main notations in Table 1. Given a positive integer d, let $[d]$ be the set $\{1,\cdots,d\}$. Given $i\in[d]$, $\mathbf{e}_i\in\mathbb{R}^d$ is the canonical vector basis with i_{th} entry being 1 and others 0. Given $(i,j,k)\in[d_1]\times[d_2]\times[d_3]$, outer product $\mathbf{e}_i\circ\mathbf{e}_j\circ\mathbf{e}_k$ is the canonical tensor basis in $\mathbb{R}^{d_1\times d_2\times d_3}$ with $(i,j,k)_{th}$ entry being 1 and others 0. For a 3-way tensor, a tube is a vector defined by fixing indices of the first two modes and varying the third one; A *slice* is a matrix defined by fixing all but two indices. Notation $\text{fft}_3(\cdot)$ denotes the *fast discrete Fourier transformation (FFT)* along *the third mode* of a 3-way tensor, i.e., MATLAB command fft($\cdot$, [], 3); similarly, $\text{ifft}_3(\cdot)$ denotes the *fast inverse discrete Fourier transformation (IFFT)* along *the third mode* of a 3-way tensor, i.e., MATLAB command ifft($\cdot$, [], 3). We use C, c and their derivatives like c', c_0, etc. to denote absolute constants, whose values may *vary from line to line*. For any $a,b\in\mathbb{R}$, let $a\vee b=\max\{a,b\}$ and $a\wedge b=\min\{a,b\}$. Let $\lceil a\rceil$ denote the closest integer to $a\in\mathbb{R}$ that is not smaller than a, and $\lfloor a\rfloor$ denotes the closest integer to $a\in\mathbb{R}$ that is not larger than a. For tensors of size $d_1\times d_2\times d_3$, we assume that $d_1\geq d_2$ without loss of generality. For simplicity, let $\tilde{d}=(d_1+d_2)d_3$. The spectral norm $\|\cdot\|$ and nuclear norm $\|\cdot\|_*$ of a matrix are defined as the maximum and the sum of its singular values, respectively. Let $\underline{\boldsymbol{0}}$ and $\underline{\boldsymbol{1}}$ denote the tensor of compatible dimension whose entries are all0/s and1/s, respectively. If the denominator is 0, we define $\frac{0}{0}=0$ in this paper, which will be used in Eqs. (38) and (39) and Eqs. (54)–(55).

2.2. Tensor singular value decomposition

Some preliminaries of tensor SVD are introduced in this subsection.

Definition 1 (T-product [13]). Let $\mathcal{T}_1\in\mathbb{R}^{d_1\times d_2\times d_3}$ and $\mathcal{T}_2\in\mathbb{R}^{d_2\times d_4\times d_3}$. The t-product of $\mathcal{T}_1$ and $\mathcal{T}_2$ is a tensor $\mathcal{T}$ of size $d_1\times d_4\times d_3$:

$$\mathcal{T}:=\mathcal{T}_1*\mathcal{T}_2, \tag{6}$$

whose $(i,j)_{th}$ tube is given by

$$\mathcal{T}(i,j,:)=\sum_{k=1}^{d_2}\mathcal{T}_1(i,k,:)\bullet\mathcal{T}_2(k,j,:),$$

where $\bullet$ denotes the circular convolution between two fibers [23].

Definition 2 (Tensor transpose [13]). Let $\mathcal{T}$ be a tensor of size $d_1\times d_2\times d_3$, then $\mathcal{T}^\top$ is the $d_2\times d_1\times d_3$ tensor obtained by transposing each of the frontal slices and then reversing the order of transposed frontal slices 2 through d_3.

Definition 3 (Identity tensor [13]). The identity tensor $\mathcal{I}\in\mathbb{R}^{d_1\times d_1\times d_3}$ is a tensor whose first frontal slice is the $d_1\times d_1$ identity matrix and all other frontal slices are zero.

Definition 4 (F-diagonal tensor [13]). A tensor is called f-diagonal if each frontal slice of the tensor is a diagonal matrix.

Definition 5 (Orthogonal tensor [13]). A tensor $\mathcal{Q}\in\mathbb{R}^{d_1\times d_1\times d_3}$ is orthogonal if $\mathcal{Q}^\top*\mathcal{Q}=\mathcal{Q}*\mathcal{Q}^\top=\mathcal{I}$.

Based on the above concepts, the tensor singular value decomposition (t-SVD) can be defined as follows. It is illustrated in Fig. 3.

Definition 6 (T-SVD, Tensor tubal-rank [13]). For any $\mathcal{T}\in\mathbb{R}^{d_1\times d_2\times d_3}$, the tensor singular value decomposition (t-SVD) of $\mathcal{T}$ is given as follows

$$\mathcal{T}=\mathcal{U}*\underline{\boldsymbol{\Lambda}}*\mathcal{V}^\top, \tag{7}$$

where $\mathcal{U}\in\mathbb{R}^{d_1\times d_1\times d_3}$, $\underline{\boldsymbol{\Lambda}}\in\mathbb{R}^{d_1\times d_2\times d_3}$, $\mathcal{V}\in\mathbb{R}^{d_2\times d_2\times d_3}$, $\mathcal{U}$ and $\mathcal{V}$ are orthogonal tensors, $\underline{\boldsymbol{\Lambda}}$ is a rectangular f-diagonal tensor.

The tensor tubal rank of $\mathcal{T}$ is defined to be the number of non-zero tubes of $\underline{\boldsymbol{\Lambda}}$ in the t-SVD factorization, i.e.,

$$r_{\text{t}}(\mathcal{T}):=\sum_i\mathbf{1}(\underline{\boldsymbol{\Lambda}}(i,i,:)\neq\mathbf{0}). \tag{8}$$

The definitions of tubal nuclear norm and tensor spectral norm will be given. The former has been applied as a convex relaxation of the tensor tubal rank in [8,19,22,25,26].

Definition 7 (Tubal nuclear norm [6,13]). For any $\mathcal{T}\in\mathbb{R}^{d_1\times d_2\times d_3}$, let $\overline{\mathbf{T}}$ denote the block-diagonal matrix of the tensor $\widetilde{\mathcal{T}}:=\text{fft}_3(\mathcal{T})$, i.e.,

$$\overline{\mathbf{T}}:=\begin{bmatrix}\widetilde{\mathbf{T}}(:,:,1) & & \\ & \ddots & \\ & & \widetilde{\mathbf{T}}(:,:,d_3)\end{bmatrix}\in\mathbb{C}^{d_1d_3\times d_2d_3}.$$

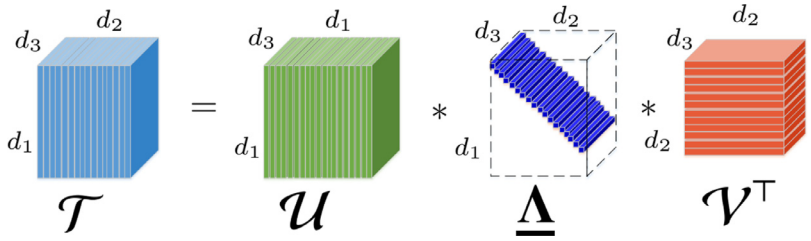


Fig. 3. Illustration of t-SVD [24].

The tubal nuclear norm $\|\mathcal{T}\|_\star$ of $\mathcal{T}$ is the rescaled matrix nuclear norm (i.e. the sum of singular values) of $\overline{\mathbf{T}}$, i.e.,

$$\|\mathcal{T}\|_\star := \frac{\|\overline{\mathbf{T}}\|_*}{d_3}. \tag{9}$$

Definition 8 (Tensor spectral norm [13])**.** The tensor spectral norm $\|\mathcal{T}\|_{\text{sp}}$ of a 3-D tensor $\mathcal{T}$ is defined as the matrix spectral norm (i.e. the largest singular value) of $\overline{\mathbf{T}}$, i.e.,

$$\|\mathcal{T}\|_{\text{sp}} := \|\overline{\mathbf{T}}\|. \tag{10}$$

It has been shown in [6] that TNN is the dual norm of tensor spectral norm.

3. The problem formulation

In this section, the t-SVD-based robust tensor decomposition will be formulated. We first introduce the observation model.

3.1. The observation model

Suppose a corrupted tensor $\mathcal{Y} \in \mathbb{R}^{d_1\times d_2\times d_3}$ is observed according to the following observation model

$$\mathcal{Y} = \mathcal{L}^* + \mathcal{S}^* + \mathcal{E}, \tag{11}$$

where $\mathcal{L}^* \in \mathbb{R}^{d_1\times d_2\times d_3}$ is the true but unknown signal tensor, $\mathcal{S}^* \in \mathbb{R}^{d_1\times d_2\times d_3}$ represents an outlier tensor with some sparsity structure, and $\mathcal{E} \in \mathbb{R}^{d_1\times d_2\times d_3}$ denotes a (deterministic or random) noise tensor.

In this paper, we assume that the signal tensor $\mathcal{L}^*$ has low tubal rank, i.e.,

$$r_t(\mathcal{L}^*) \ll \min\{d_1, d_2\}. \tag{12}$$

Besides, we also assume $\mathcal{S}^*$ with support Θ_s satisfies *one of the three sparsity settings*:

- Setting 1. $\mathcal{S}^*$ has element-wise sparsity, i.e., its support $\Theta_s \subset [d_1]\times[d_2]\times[d_3]$ satisfies $|\Theta_s| \ll d_1d_2d_3$. Then, $\mathcal{S}^*$ can represent element-wisely sparse outliers (see Fig. 2-a). When the noise tensor $\mathcal{E} = \mathbf{0}$, Eq. (11) is the observation model of TRPCA [19].
- Setting 2. $\mathcal{S}^*$ has tube-wise sparsity, i.e., its support Θ_s satisfies $\Theta_s \subset \Theta_t \times [d_3]$ with $\Theta_t \subset [d_1]\times[d_2]$ and $|\Theta_t| \ll d_1d_2$. Then, $\mathcal{S}^*$ can represent tube-wisely sparse outliers (see Fig. 2-b). When the noise tensor $\mathcal{E} = \mathbf{0}$, Eq. (11) is the observation model of TRPCA with tube corruption [22].
- Setting 3. $\mathcal{S}^*$ has lateral-slice-wise sparsity, i.e., its support $\Theta_s \subset [d_1]\times\Theta_{ls}\times[d_3]$ with $\Theta_{ls} \subset [d_2]$ and $|\Theta_{ls}| \ll d_2$. Then, $\mathcal{S}^*$ can represent lateral-slice-wise sparse sample outliers (see Fig. 2-c). When the noise tensor $\mathcal{E} = \mathbf{0}$, Eq. (11) is the observation model of outlier robust tensor PCA (OR-TPCA) [8].

The goal of RTD is to recover $\mathcal{L}^*$ and $\mathcal{S}^*$ from the corrupted observation $\mathcal{Y}$ satisfying the observation model (11). Considering the observation model in Eq. (11), we make the following assumption on the true signal tensor $\mathcal{L}^*$ to avoid ambiguity in the decomposition to some extent.

Assumption 1. The l_∞-norm of $\mathcal{L}^*$ (i.e., the maximum of entry-wise absolute value) is upper bounded by a known constant α, i.e.,

$$\|\mathcal{L}^*\|_{l_\infty} \le \alpha.$$

Remark 1. We have the following remarks on Assumption 1:

(I). The l_∞-norm boundedness is a natural assumption in many real applications. For example, the magnitude of the true image or video tensor is bounded by 255 in image or video restoration.

(II). The l_∞-norm boundedness is milder than the tensor incoherent conditions (TICs) proposed in [8,19] for TRPCA and OR-TPCA in noiseless settings. It is also used in noisy/robust matrix completion [27,28] and noisy tensor completion [24,25]. It serves as a relaxation of the non-spiky condition adopted in robust matrix decomposition [29].

3.2. The proposed estimator

Given a corrupted observation $\mathcal{Y}$, a penalized least squares estimator is defined to estimate $\mathcal{L}^*$ and $\mathcal{S}^*$ as follows:

$$(\hat{\mathcal{L}}, \hat{\mathcal{S}}) = \underset{\mathcal{L},\mathcal{S}}{\text{argmin}} \frac{1}{2}\|\mathcal{L}+\mathcal{S}-\mathcal{Y}\|_{\text{F}}^2 + \lambda\|\mathcal{L}\|_\star + \mu R(\mathcal{S}), \quad \text{s.t.}\|\mathcal{L}\|_{l_\infty} \le \alpha, \tag{13}$$

where λ and μ are positive regularization parameters, $R(\cdot)$ is a regularizer to impose certain sparsity in the final solution $\hat{\mathcal{S}}$. For $\mathcal{S}^*$ being element-wisely, tube-wisely or slice-wisely sparse, we choose $R(\cdot)$ as $\|\cdot\|_{l_1}$, $\|\cdot\|_{\text{tube}_1}$ or $\|\cdot\|_{\text{slice}_1}$, respectively.

4. Statistical performance: Minimax near-optimal error bounds

In this section, statistical performance of the proposed estimator $(\hat{\mathcal{L}}, \hat{\mathcal{S}})$ in Eq. (13) will be analyzed. Specifically, we first derive upper bounds on the estimation error both deterministically and non-asymptotically, and then establish lower bounds on the error in a minimax sense.

4.1. Upper bounds on the estimation error

For the ease of notation, we use $\Delta_{\mathcal{L}} = \hat{\mathcal{L}} - \mathcal{L}^*$ and $\Delta_{\mathcal{S}} = \hat{\mathcal{S}} - \mathcal{S}^*$ to denote the error tensors of $\mathcal{L}^*$ and $\mathcal{S}^*$, respectively. To explore the statistical performance of the estimator $(\hat{\mathcal{L}}, \hat{\mathcal{S}})$, we will give upper bounds on the sum of squared Frobenius norms $\|\Delta_{\mathcal{L}}\|_{\text{F}}^2 + \|\Delta_{\mathcal{S}}\|_{\text{F}}^2$.

4.1.1. Deterministic bounds

When tensor $\mathcal{E}$ in the observation model (11) represents any deterministic or random noise, we derive upper bounds on the estimation error in a deterministic sense.

To bound the error, we need Lemma 1 at first. For the ease of notation, let $R^*(\cdot)$ denote the dual norm of $R(\cdot)$, which is chosen as $\|\cdot\|_{l_\infty}$, $\|\cdot\|_{\text{tube}_\infty}$, or $\|\cdot\|_{\text{slice}_\infty}$ for $R(\cdot)$ being $\|\cdot\|_{l_1}$, $\|\cdot\|_{\text{tube}_1}$, or $\|\cdot\|_{\text{slice}_1}$ respectively.

Lemma 1. *Choose $\lambda \geq 2\|\mathcal{E}\|_{\text{sp}}$ in Problem (13) and $\mu \geq 2(R^*(\mathcal{E}) + 2\alpha R^*(\mathbf{1}))$. Then, there exist decompositions $\Delta_{\mathcal{L}} = \Delta'_{\mathcal{L}} + \Delta''_{\mathcal{L}}$ and $\Delta_{\mathcal{S}} = \Delta'_{\mathcal{S}} + \Delta''_{\mathcal{S}}$, such that*

(I). *a rank inequality holds: $r_{\text{t}}(\Delta'_{\mathcal{L}}) \leq 2r_{\text{t}}(\mathcal{L}^*)$, and*
(II). *a norm inequality holds:*

$$\lambda\|\Delta''_{\mathcal{L}}\|_\star + \mu R(\Delta''_{\mathcal{S}}) \leq 3\Big(\lambda\|\Delta'_{\mathcal{L}}\|_\star + \mu R(\Delta'_{\mathcal{S}})\Big). \tag{14}$$

The proof can be found in the supplemental material. Based on the lemma, we are able to establish the deterministic bounds on the estimation error in the following theorem.

Theorem 1. *Choose $\lambda \geq 2\|\mathcal{E}\|_{\text{sp}}$ in Problem (13). Then the following statements hold[1]:*

(I). *If $R(\cdot) = \|\cdot\|_{l_1}$, by setting $\mu \geq 2(\|\mathcal{E}\|_{l_\infty} + 2\alpha)$, we have*

$$\|\Delta_{\mathcal{L}}\|_{\text{F}}^2 + \|\Delta_{\mathcal{S}}\|_{\text{F}}^2 \leq 18\lambda^2 r_{\text{t}}(\mathcal{L}^*) + 9\mu^2\|\mathcal{S}^*\|_{l_0}. \tag{15}$$

(II). *If $R(\cdot) = \|\cdot\|_{\text{tube}_1}$, by setting $\mu \geq 2(\|\mathcal{E}\|_{\text{tube}_\infty} + 2\alpha\sqrt{d_3})$, we have*

$$\|\Delta_{\mathcal{L}}\|_{\text{F}}^2 + \|\Delta_{\mathcal{S}}\|_{\text{F}}^2 \leq 18\lambda^2 r_{\text{t}}(\mathcal{L}^*) + 9\mu^2\|\mathcal{S}^*\|_{\text{tube}_0}. \tag{16}$$

(III). *If $R(\cdot) = \|\cdot\|_{\text{slice}_1}$, by setting $\mu \geq 2(\|\mathcal{E}\|_{\text{slice}_\infty} + 2\alpha\sqrt{d_1 d_3})$, we have*

$$\|\Delta_{\mathcal{L}}\|_{\text{F}}^2 + \|\Delta_{\mathcal{S}}\|_{\text{F}}^2 \leq 18\lambda^2 r_{\text{t}}(\mathcal{L}^*) + 9\mu^2\|\mathcal{S}^*\|_{\text{slice}_0}. \tag{17}$$

The proof is given in the supplemental material. We can see from Theorem 1 that the upper bounds have linear scaling behavior with the tubal rank of $\mathcal{L}^*$ and the sparsity of $\mathcal{S}^*$, when the regularization parameters λ and μ exceed certain values. When $d_3 = 1$, the upper bounds are consistent with the bounds in [29].

4.1.2. Non-asymptotic bounds

When the elements of $\mathcal{E} \in \mathbb{R}^{d_1\times d_2\times d_3}$ follow independent and identically distributed (*i.i.d.*) Gaussian distribution $\mathcal{N}(0, \sigma^2)$, we will give non-asymptotic upper bounds on the estimation error. To this end, we need the following lemmas whose proofs can be found in the supplemental material.

Lemma 2. *If the elements of $\mathcal{G} \in \mathbb{R}^{d_1\times d_2\times d_3}$ follow i.i.d. Gaussian distribution $\mathcal{N}(0, 1)$, then it holds that*

$$\mathbb{P}\Big[\|\mathcal{G}\|_{\text{sp}} \geq 2(\sqrt{d_1} + \sqrt{d_2})\sqrt{d_3}\Big] \leq e^{-c(\sqrt{d_1}+\sqrt{d_2})^2}. \tag{18}$$

Lemma 3. *If the elements of $\mathcal{G} \in \mathbb{R}^{d_1\times d_2\times d_3}$ follow i.i.d. Gaussian distribution $\mathcal{N}(0, 1)$, then we have the following relationships.*

(I). *The l_∞-norm of $\mathcal{G}$ satisfy the probability inequality*

$$\mathbb{P}\Big[\|\mathcal{G}\|_{l_\infty} \geq 2\sqrt{\log(d_1 d_2 d_3)}\Big] \leq \frac{1}{d_1 d_2 d_3}. \tag{19}$$

(II). *The tube$_\infty$-norm of $\mathcal{G}$ satisfy the probability inequality*

$$\mathbb{P}\Big[\|\mathcal{G}\|_{\text{tube}_\infty} \geq \sqrt{d_3} + 3\sqrt{\log(d_1 d_2)}\Big] \leq \frac{1}{d_1 d_2}. \tag{20}$$

(III). *The slice$_\infty$-norm of $\mathcal{G}$ satisfy the probability inequality*

$$\mathbb{P}\Big[\|\mathcal{G}\|_{\text{slice}_\infty} \geq \sqrt{d_1 d_3} + 3\sqrt{\log d_2}\Big] \leq \frac{1}{d_2}. \tag{21}$$

[1] The relevant tensor norms are defined in Table 1.

Based on Theorem 1 and Lemmas 2–3, we are in a position to upper bound the estimation error. Before showing the error bounds under *i.i.d.* Gaussian noise, we define the low rank ratio $\boldsymbol{\varrho}_{\text{r}} \in [0, 1]$ of $\mathcal{L}^*$ and the sparsity ratio $\boldsymbol{\varrho}_{\text{s}} \in [0, 1]$ of $\mathcal{S}^*$ respectively as follows:

$$\boldsymbol{\varrho}_{\text{r}}(\mathcal{L}^*) := \frac{r_{\text{t}}(\mathcal{L}^*)}{d_1 \wedge d_2},\ \boldsymbol{\varrho}_{\text{s}}(\mathcal{S}^*) := \begin{cases} \frac{\|\mathcal{S}^*\|_{l_0}}{d_1 d_2 d_3}, & \text{element-wise sparsity,} \\ \frac{\|\mathcal{S}^*\|_{\text{tube}_0}}{d_1 d_2}, & \text{tube-wise sparsity,} \\ \frac{\|\mathcal{S}^*\|_{\text{slice}_0}}{d_2}, & \text{slice-wise sparsity.} \end{cases} \tag{22}$$

Thus, the more complex the signal tensor $\mathcal{L}^*$, the higher $\boldsymbol{\varrho}_{\text{r}}(\mathcal{L}^*)$; the heavier the outliers $\mathcal{S}^*$, the higher $\boldsymbol{\varrho}_{\text{s}}(\mathcal{S}^*)$.

Theorem 2. *Consider the case where the elements of $\mathcal{E} \in \mathbb{R}^{d_1\times d_2\times d_3}$ follow i.i.d. Gaussian distribution $\mathcal{N}(0, \sigma^2)$. Choose $\lambda = 4\sigma(\sqrt{d_1} + \sqrt{d_2})\sqrt{d_3}$ in Problem (13). Then the following statements hold:*

(I). *If $R(\cdot) = \|\cdot\|_{l_1}$, by setting $\mu = 4\sigma\sqrt{\log(d_1 d_2 d_3)} + 8\alpha$, we have*

$$\frac{\|\Delta_{\mathcal{L}}\|_{\text{F}}^2 + \|\Delta_{\mathcal{S}}\|_{\text{F}}^2}{d_1 d_2 d_3} \leq 1152\sigma^2\boldsymbol{\varrho}_{\text{r}}(\mathcal{L}^*) + 288\Big(\sigma^2\log(d_1 d_2 d_3) + \alpha^2\Big)\boldsymbol{\varrho}_{\text{s}}(\mathcal{S}^*), \tag{23}$$

with probability at least $1 - \exp(-c(\sqrt{d_1} + \sqrt{d_2})^2) - (d_1 d_2 d_3)^{-1}$.

(II). *If $R(\cdot) = \|\cdot\|_{\text{tube}_1}$ and $d_3 \gg \log(d_1 d_2)$, by setting $\mu = 2(\sigma\sqrt{d_3} + 3\sigma\sqrt{\log(d_1 d_2)} + 2\alpha\sqrt{d_3})$, we have*

$$\frac{\|\Delta_{\mathcal{L}}\|_{\text{F}}^2 + \|\Delta_{\mathcal{S}}\|_{\text{F}}^2}{d_1 d_2 d_3} \leq 1152\sigma^2\boldsymbol{\varrho}_{\text{r}}(\mathcal{L}^*) + 576(\sigma \vee \alpha)^2\boldsymbol{\varrho}_{\text{s}}(\mathcal{S}^*), \tag{24}$$

with probability at least $1 - \exp(-c(\sqrt{d_1} + \sqrt{d_2})^2) - (d_1 d_2)^{-1}$.

(III). *If $R(\cdot) = \|\cdot\|_{\text{slice}_1}$ and $d_1 d_3 \gg \log(d_2)$, by setting $\mu = 2(\sigma\sqrt{d_1 d_3} + 3\sigma\sqrt{\log d_2} + 2\alpha\sqrt{d_1 d_3})$, we have*

$$\frac{\|\Delta_{\mathcal{L}}\|_{\text{F}}^2 + \|\Delta_{\mathcal{S}}\|_{\text{F}}^2}{d_1 d_2 d_3} \leq 1152\sigma^2\boldsymbol{\varrho}_{\text{r}}(\mathcal{L}^*) + 576(\sigma \vee \alpha)^2\boldsymbol{\varrho}_{\text{s}}(\mathcal{S}^*), \tag{25}$$

with probability at least $1 - \exp(-c(\sqrt{d_1} + \sqrt{d_2})^2) - d_2^{-1}$.

According to Theorem 2, the bounds in Eqs. (23)-(25) can be summarized uniformly as follows

$$\frac{\|\Delta_{\mathcal{L}}\|_{\text{F}}^2 + \|\Delta_{\mathcal{S}}\|_{\text{F}}^2}{d_1 d_2 d_3} \leq c_1\sigma^2\boldsymbol{\varrho}_{\text{r}}(\mathcal{L}^*) + c_2(\varsigma \vee \alpha)^2\boldsymbol{\varrho}_{\text{s}}(\mathcal{S}^*), \tag{26}$$

where $\varsigma = \sigma\log(d_1 d_2 d_3)$ for element-wisely sparse $\mathcal{S}^*$, and $\varsigma = \sigma$ for tube-wisely or slice-wisely sparse $\mathcal{S}^*$. It is notable that Eq. (26) is consistent with our intuition: the more complex the signal tensor (i.e., higher $\boldsymbol{\varrho}_{\text{r}}(\mathcal{L}^*)$), the heavier the outliers (i.e., higher $\boldsymbol{\varrho}_{\text{s}}(\mathcal{S}^*)$), and the heavier the noise (i.e., larger σ), the larger the estimation error will be.

To the best of our knowledge, Theorem 2 for the first time establishes error bounds for robust tensor decomposition when the underlying tensor is low-tubal-rank. The proposed upper bounds in Eqs. (23)-(25) will be shown to be near-optimal in the minimax sense in the next subsection. The comparison with previous works are shown in the following remarks.

Remark 2 (Difference from SNN-based robust tensor decomposition (SNN-RTD) [5])**.** SNN-RTD models the underlying tensor as low-Tucker-rank, and this paper assumes the underlying tensor to

be low-tubal-rank. Besides, the proposed upper bounds are minimax near-optimal, whereas the bound in SNN-RTD is not.

Remark 3 (Degeneration to robust matrix decomposition [29]). When $d_3 = 1$, the robust tensor decomposition degenerates to robust matrix decomposition, and the degenerated bounds in Eqs. (23) and (25) are consistent with the bounds shown in Corollaries 2 and 5 in [29], respectively.

Remark 4 (No exact recovery guarantee). When the noise $\mathcal{E}$ vanishes, i.e., $\sigma = 0$, the estimation error is upper bounded by $C\alpha^2\boldsymbol{\varrho}_s$ which is not 0. That is, the theory in this paper cannot guarantee exact recovery of $\mathcal{L}^*$ and $\mathcal{S}^*$.

This differs from the theoretic analysis of TRPCA [19] and ORTPCA [8] which assumes the underlying tensor $\mathcal{L}^*$ satisfies some tensor incoherence conditions (TICs) and can guarantee exact recovery of $\mathcal{L}^*$ and $\mathcal{S}^*$. The difference lies in the fact that this paper adopts "the l_∞-norm-boundedness assumption" on the true tensor $\mathcal{L}^*$ — a much weaker assumption than TICs in [19] and [8]. TICs inherently ensure that the low-rank tensor $\mathcal{L}^*$ is not sparse, whereas "the l_∞-norm-boundedness assumption" cannot. For the comparison of TICs and l_∞-norm-boundedness, please refer to [24].

4.2. Minimax lower bounds

In Section 4.1.2, we establish upper bounds on the estimation error for *i.i.d.* Gaussian noise. Then one may ask the complementary questions: how tight are these upper bounds? Are there fundamental (algorithm-independent) limits of estimation error in robust tensor decomposition? In this section, we will answer the questions.

Consider the case where the elements of $\mathcal{E} \in \mathbb{R}^{d_1\times d_2\times d_3}$ follow *i.i.d.* Gaussian distribution $\mathcal{N}(0,\sigma^2)$ with known $\sigma > 0$. Given some class $\mathbb{A}$ of tensors, we define the associated element-wise minimax error as follows

$$\mathsf{M}(\mathbb{A}) := \inf_{(\hat{\mathcal{L}},\hat{\mathcal{S}})} \sup_{(\mathcal{L}^*,\mathcal{S}^*)\in\mathbb{A}} \mathbb{E}\left[\frac{\|\hat{\mathcal{L}}-\mathcal{L}^*\|_{\mathrm{F}}^2 + \|\hat{\mathcal{S}}-\mathcal{S}^*\|_{\mathrm{F}}^2}{d_1d_2d_3}\right], \tag{27}$$

where the infimum ranges over all pairs of estimators $(\hat{\mathcal{L}},\hat{\mathcal{S}})$, the supremum ranges over all pairs of "true" tensors $(\mathcal{L}^*,\mathcal{S}^*)$ in the given tensor class $\mathbb{A}$, and the expectation is taken over the *i.i.d.* Gaussian noises. We come up with the following theorem.

Theorem 3. *Consider the case where the elements of $\mathcal{E} \in \mathbb{R}^{d_1\times d_2\times d_3}$ follow i.i.d. Gaussian distribution $\mathcal{N}(0,\sigma^2)$, where σ is known. Then for $1 \le r \le \min\{d_1, d_2\}$, the following statements hold with positive constants c_i', c_i'' and $\beta_i \in (0,1)$, $i = 1,2,3$:*

(I). *For any positive integer $s \le d_1d_2d_3/2$, let $\phi_{\mathrm{e}} := (\sigma\wedge\alpha)^2\big(c_1'r/(d_1\wedge d_2) + c_1''s/(d_1d_2d_3)\big)$, and define the class of tensors*

$$\mathbb{A}_{\mathrm{e}}(r,s,\alpha) := \Big\{(\mathcal{L},\mathcal{S})\Big|\mathrm{r_t}(\mathcal{L}) \le r, \|\mathcal{L}\|_{l_\infty} \le \alpha, \|\mathcal{S}\|_{l_0} \le s\Big\}.$$

Then it holds that

$$\mathsf{M}(\mathbb{A}_{\mathrm{e}}(r,s,\alpha)) \ge \beta_1\phi_{\mathrm{e}}. \tag{28}$$

(II). *For any positive integer $s \le d_1d_2/2$, let $\phi_{\mathrm{t}} := (\sigma\wedge\alpha)^2\big(c_2'r/(d_1\wedge d_2) + c_2''s/(d_1d_2)\big)$ and define the class of tensors*

$$\mathbb{A}_{\mathrm{t}}(r,s,\alpha) := \Big\{(\mathcal{L},\mathcal{S})\Big|\mathrm{r_t}(\mathcal{L}) \le r, \|\mathcal{L}\|_{l_\infty} \le \alpha, \|\mathcal{S}\|_{\mathrm{tube}_0} \le s\Big\}.$$

Then it holds that

$$\mathsf{M}(\mathbb{A}_{\mathrm{t}}(r,s,\alpha)) \ge \beta_2\phi_{\mathrm{t}}. \tag{29}$$

(III). *For any positive integer $s \le d_2/2$, let $\phi_{\mathrm{s}} := (\sigma\wedge\alpha)^2\big(c_3'r/(d_1\wedge d_2) + c_3''s/d_2\big)$, and define the class of tensors*

$$\mathbb{A}_{\mathrm{s}}(r,s,\alpha) := \Big\{(\mathcal{L},\mathcal{S})\Big|\mathrm{r_t}(\mathcal{L}) \le r, \|\mathcal{L}\|_{l_\infty} \le \alpha, \|\mathcal{S}\|_{\mathrm{slice}_0} \le s\Big\}.$$

Then it holds that

$$\mathsf{M}(\mathbb{A}_{\mathrm{s}}(r,s,\alpha)) \ge \beta_3\phi_{\mathrm{s}}. \tag{30}$$

In Theorem 3, for some certain classes of $(\mathcal{L}^*,\mathcal{S}^*)$, Eqs. (28)-(30) establish minimax lower bounds on the estimation errors for element-wisely, tube-wisely, and slice-wisely sparse outliers, respectively. When σ and α are known constants, the minimax lower bounds in Eqs. (28)-(30) can be unified as

$$\inf_{(\hat{\mathcal{L}},\hat{\mathcal{S}})} \sup_{(\mathcal{L}^*,\mathcal{S}^*)} \mathbb{E}\left[\frac{\|\Delta_{\mathcal{L}}\|_{\mathrm{F}}^2 + \|\Delta_{\mathcal{S}}\|_{\mathrm{F}}^2}{d_1d_2d_3}\right] \ge c'(\sigma\wedge\alpha)^2\boldsymbol{\varrho}_{\mathrm{r}}(\mathcal{L}^*) + c''(\sigma\wedge\alpha)^2\boldsymbol{\varrho}_{\mathrm{s}}(\mathcal{S}^*), \tag{31}$$

for some $(\mathcal{L}^*,\mathcal{S}^*)$ in certain tensor classes, where $\boldsymbol{\varrho}_{\mathrm{r}}(\mathcal{L}^*)$ and $\boldsymbol{\varrho}_{\mathrm{s}}(\mathcal{S}^*)$ denote the low-rank ratio and sparse ratio defined in Eq. (22). Comparing Eqs. (26) and (31), the proposed upper bounds in Theorem 2 are minimax optimal (up to a logarithm factor in the setting of element-wise outliers or constant factors for tube-wise and slice-wise outliers). That is, no estimator can provide better estimations (up to a logarithm factor or constant factor) in the minimax sense than the proposed estimator.

5. Optimization algorithms

5.1. An ADMM optimizer

We first propose an algorithm based on the alternating direction method of multipliers (ADMM) to solve the proposed estimator. By introducing auxiliary $\mathcal{K},\mathcal{M},\mathcal{T}$, we get

$$\begin{aligned}\min_{\substack{\mathcal{L},\mathcal{S},\\ \mathcal{K},\mathcal{T},\mathcal{M}}}\quad & \tfrac{1}{2}\|\mathcal{L}+\mathcal{S}-\mathcal{Y}\|_{\mathrm{F}}^2 + \lambda\|\mathcal{K}\|_\star + \mu R(\mathcal{T})\\ \text{s.t.}\quad & \mathcal{K}=\mathcal{L},\ \mathcal{T}=\mathcal{S},\ \mathcal{M}=\mathcal{L},\ \|\mathcal{M}\|_{l_\infty}\le\alpha.\end{aligned} \tag{32}$$

The augmented Lagrangian of Problem (32) is as follows:

$$\begin{aligned}&L_\rho(\mathcal{L},\mathcal{S},\mathcal{K},\mathcal{T},\mathcal{M},\mathcal{Y}_1,\mathcal{Y}_2,\mathcal{Y}_3)\\ &= \frac{1}{2}\|\mathcal{L}+\mathcal{S}-\mathcal{Y}\|_{\mathrm{F}}^2 + \lambda\|\mathcal{K}\|_\star + \mu R(\mathcal{T}) + \delta(\mathcal{M})\\ &\quad + \langle\mathcal{Y}_1,\mathcal{K}-\mathcal{L}\rangle + \frac{\rho}{2}\|\mathcal{K}-\mathcal{L}\|_{\mathrm{F}}^2 + \langle\mathcal{Y}_2,\mathcal{T}-\mathcal{S}\rangle\\ &\quad + \frac{\rho}{2}\|\mathcal{T}-\mathcal{S}\|_{\mathrm{F}}^2 + \langle\mathcal{Y}_3,\mathcal{M}-\mathcal{L}\rangle + \frac{\rho}{2}\|\mathcal{M}-\mathcal{L}\|_{\mathrm{F}}^2,\end{aligned} \tag{33}$$

where $\rho > 0$ is a penalty parameter, and $\mathcal{Y}_i \in \mathbb{R}^{d_1\times d_2\times d_3}, i \le 3$ are Lagrangian multipliers.

Then, we update the variables alternatively by fixing others. The details are shown as follows.

- Update $(\mathcal{L},\mathcal{S})$: we update $(\mathcal{L},\mathcal{S})$ simultaneously as follows:

$$\begin{aligned}(\mathcal{L}^{t+1},\mathcal{S}^{t+1}) &= \operatorname*{argmin}_{\mathcal{L},\mathcal{S}} L_\rho(\mathcal{L},\mathcal{S},\mathcal{K}^t,\mathcal{T}^t,\mathcal{M}^t,\mathcal{Y}_1^t,\mathcal{Y}_2^t,\mathcal{Y}_3^t)\\ &= \operatorname*{argmin}_{\mathcal{L},\mathcal{S}} \frac{1}{2}\|\mathcal{L}+\mathcal{S}-\mathcal{Y}\|_{\mathrm{F}}^2 + \langle\mathcal{Y}_1^t,\mathcal{K}^t-\mathcal{L}\rangle + \frac{\rho}{2}\|\mathcal{K}^t-\mathcal{L}\|_{\mathrm{F}}^2\\ &\quad + \langle\mathcal{Y}_2^t,\mathcal{T}^t-\mathcal{S}\rangle + \frac{\rho}{2}\|\mathcal{T}^t-\mathcal{S}\|_{\mathrm{F}}^2 + \langle\mathcal{Y}_3^t,\mathcal{M}^t-\mathcal{L}\rangle + \frac{\rho}{2}\|\mathcal{M}^t-\mathcal{L}\|_{\mathrm{F}}^2.\end{aligned}$$

Taking derivatives with respective to $\mathcal{L}$ and $\mathcal{S}$ and set them to zero, we obtain

$$\begin{aligned}&\mathcal{L}+\mathcal{S}-\mathcal{Y}-\mathcal{Y}_1^t+\rho(\mathcal{L}-\mathcal{K}^t)-\mathcal{Y}_3^t+\rho(\mathcal{L}-\mathcal{M}^t) = \underline{\mathbf{0}},\\ &\mathcal{L}+\mathcal{S}-\mathcal{Y}-\mathcal{Y}_2^t+\rho(\mathcal{S}-\mathcal{T}^t) = \underline{\mathbf{0}}.\end{aligned}$$

Then, we have

$$\mathcal{L}^{t+1}=\frac{(\rho+1)\mathcal{A}-\mathcal{B}+\rho\mathcal{Y}}{\rho(2\rho+3)},\quad \mathcal{S}^{t+1}=\frac{(2\rho+1)\mathcal{B}-\mathcal{A}+2\rho\mathcal{Y}}{\rho(2\rho+3)}, \tag{34}$$

where $\mathcal{A}=\rho\mathcal{K}^t+\mathcal{Y}_1^t+\rho\mathcal{M}^t+\mathcal{Y}_3^t$ and $\mathcal{B}=\rho\mathcal{T}^t+\mathcal{Y}_2^t$.

- Update $\mathcal{K}$: we update $\mathcal{K}$ as follows:

$$\begin{aligned}\mathcal{K}^{t+1}&=\operatorname{argmin}_{\mathcal{K}} L_\rho(\mathcal{L}^{t+1},\mathcal{S}^{t+1},\mathcal{K},\mathcal{T}^t,\mathcal{M}^t,\mathcal{Y}_1^t,\mathcal{Y}_2^t,\mathcal{Y}_3^t)\\&=\operatorname{argmin}_{\mathcal{K}}\lambda\|\mathcal{K}\|_\star+\langle\mathcal{Y}_1^t,\mathcal{K}-\mathcal{L}^{t+1}\rangle+\tfrac{\rho}{2}\|\mathcal{K}-\mathcal{L}^{t+1}\|_{\mathrm{F}}^2\\&=\operatorname{Prox}_{\lambda/\rho}^{\|\cdot\|_\star}(\mathcal{L}^{t+1}-\mathcal{Y}_1^t/\rho)\end{aligned} \tag{35}$$

where $\operatorname{Prox}_\tau^{\|\cdot\|_\star}(\cdot)$ is the proximal operator of tubal nuclear norm. In [30], a closed-form expression of $\operatorname{Prox}_\tau^{\|\cdot\|_\star}(\cdot)$ is given as follows:

Lemma 4 [30]. *For any 3-way tensor $\mathcal{A}\in\mathbb{R}^{d_1\times d_2\times d_3}$ with reduced t-SVD $\mathcal{A}=\mathcal{U}*\underline{\mathbf{\Lambda}}*\mathcal{V}^\top$, where $\mathcal{U}\in\mathbb{R}^{d_1\times r\times d_3}$ and $\mathcal{V}\in\mathbb{R}^{d_2\times r\times d_3}$ are orthogonal tensors and $\underline{\mathbf{\Lambda}}\in\mathbb{R}^{r\times r\times d_3}$ is the f-diagonal tensor of singular tubes, the proximal operator $\operatorname{Prox}_\tau^{\|\cdot\|_\star}(\cdot)$ at $\mathcal{A}$ can be computed by*[2]*:*

$$\begin{aligned}\operatorname{Prox}_\tau^{\|\cdot\|_\star}(\mathcal{A}):&=\operatorname*{argmin}_{\mathcal{X}}\tau\|\mathcal{X}\|_\star+\frac{1}{2}\|\mathcal{X}-\mathcal{A}\|_{\mathrm{F}}^2\\&=\mathcal{U}*\operatorname{ifft}_3(\max(\operatorname{fft}_3(\underline{\mathbf{\Lambda}})-\tau,0))*\mathcal{V}^\top.\end{aligned} \tag{36}$$

- Update $\mathcal{T}$: we update $\mathcal{T}$ as follows

$$\begin{aligned}\mathcal{T}^{t+1}&=\operatorname*{argmin}_{\mathcal{T}} L_\rho(\mathcal{L}^{t+1},\mathcal{S}^{t+1},\mathcal{K}^{t+1},\mathcal{T},\mathcal{M}^t,\mathcal{Y}_1^t,\mathcal{Y}_2^t,\mathcal{Y}_3^t)\\&=\operatorname*{argmin}_{\mathcal{T}}\mu R(\mathcal{T})+\langle\mathcal{Y}_2^t,\mathcal{T}-\mathcal{S}^{t+1}\rangle+\frac{\rho}{2}\|\mathcal{T}-\mathcal{S}^{t+1}\|_{\mathrm{F}}^2\\&=\operatorname{Prox}_{\mu/\rho}^{R(\cdot)}(\mathcal{S}^{t+1}-\mathcal{Y}_2^t/\rho)\end{aligned} \tag{37}$$

where $\operatorname{Prox}_\tau^{R(\cdot)}(\cdot)$ is the proximal operator of $R(\cdot)$ which can be computed as follows.

(I). When $R(\cdot)=\|\cdot\|_{l_1}$, the proximal operator is the well known soft thresholding operator explicitly given as follows [6]

$$\begin{aligned}\operatorname{Prox}_\tau^{\|\cdot\|_{l_1}}(\mathcal{A}):&=\operatorname*{argmin}_{\mathcal{X}}\tau\|\mathcal{X}\|_{l_1}+\frac{1}{2}\|\mathcal{X}-\mathcal{A}\|_{\mathrm{F}}^2\\&=\operatorname{sign}(\mathcal{A})\circledast\big(|\mathcal{A}|-\tau,0\big)_+,\end{aligned}$$

where $\circledast$*ledast* denotes the element-wise tensor product.

(II). When $R(\cdot)=\|\cdot\|_{\mathrm{tube}_1}$, its proximal operator is the soft thresholding operator on tubes with closed-form solution [22]

$$\operatorname{Prox}_\tau^{\|\cdot\|_{\mathrm{tube}_1}}(\mathcal{A}):=\operatorname*{argmin}_{\mathcal{X}}\tau\|\mathcal{X}\|_{\mathrm{tube}_1}+\frac{1}{2}\|\mathcal{X}-\mathcal{A}\|_{\mathrm{F}}^2=\mathcal{B}, \tag{38}$$

where $\mathcal{B}(i,j,:)=\mathcal{A}(i,j,:)(1-\tau/\|\mathcal{A}(i,j,:)\|_2)_+$[3], for all $(i,j)\in[d_1]\times[d_2]$.

(III). When $R(\cdot)=\|\cdot\|_{\mathrm{slice}_1}$, the proximal operator is the soft thresholding operator on slices whose closed-form solution is given as [8]

$$\operatorname{Prox}_\tau^{\|\cdot\|_{\mathrm{slice}_1}}(\mathcal{A}):=\operatorname*{argmin}_{\mathcal{X}}\tau\|\mathcal{X}\|_{\mathrm{slice}_1}+\frac{1}{2}\|\mathcal{X}-\mathcal{A}\|_{\mathrm{F}}^2=\mathcal{B}, \tag{39}$$

where $\mathcal{B}(:,j,:)=\mathcal{A}(:,j,:)(1-\tau/\|\mathcal{A}(:,j,:)\|_{\mathrm{F}})_+$, for all $j\in[d_2]$.

- Update $\mathcal{M}$: we update $\mathcal{M}$ in the following manner:

$$\begin{aligned}\mathcal{M}^{t+1}&=\operatorname*{argmin}_{\mathcal{M}} L_\rho(\mathcal{L}^{t+1},\mathcal{S}^{t+1},\mathcal{K}^{t+1},\mathcal{T}^{t+1},\mathcal{M},\mathcal{Y}_1^t,\mathcal{Y}_2^t,\mathcal{Y}_3^t)\\&=\operatorname*{argmin}_{\mathcal{M}}\delta(\mathcal{M})+\langle\mathcal{Y}_3^t,\mathcal{M}-\mathcal{L}^{t+1}\rangle+\frac{\rho}{2}\|\mathcal{M}-\mathcal{L}^{t+1}\|_{\mathrm{F}}^2\\&=\operatorname{Proj}_\alpha^{\|\cdot\|_{l_\infty}}(\mathcal{L}^{t+1}-\mathcal{Y}_3^t/\rho),\end{aligned} \tag{40}$$

where $\operatorname{Proj}_\alpha^{\|\cdot\|_{l_\infty}}(\cdot)$ is a projection into the l_∞-norm ball of radius α serving as a clipping operator with a closed-form solution given as follows [25]:

$$\operatorname{Proj}_\alpha^{\|\cdot\|_{l_\infty}}(\mathcal{A})=\operatorname{sign}(\mathcal{A})\circledast\min\{|\mathcal{A}|,\alpha\}.$$

- Update $\mathcal{Y}_1,\mathcal{Y}_2,\mathcal{Y}_3$: the dual variables are updated by

$$\begin{aligned}\mathcal{Y}_1^{t+1}&=\mathcal{Y}_1^t+\rho(\mathcal{K}^{t+1}-\mathcal{L}^{t+1}),\\\mathcal{Y}_2^{t+1}&=\mathcal{Y}_2^t+\rho(\mathcal{T}^{t+1}-\mathcal{S}^{t+1}),\\\mathcal{Y}_3^{t+1}&=\mathcal{Y}_2^t+\rho(\mathcal{M}^{t+1}-\mathcal{L}^{t+1}).\end{aligned} \tag{41}$$

The algorithm is summarized in Algorithm 1.

Algorithm 1 Solve Problem (32) by ADMM.

Require: Observation $\mathcal{Y}$, parameters λ,μ,α and ρ.

1: Initialize $\mathcal{L}^0=\mathcal{S}^0=\mathcal{K}^0=\mathcal{T}^0=\mathcal{M}^0=\mathbf{0}$, $\mathcal{Y}_1^0=\mathcal{Y}_2^0=\mathcal{Y}_3^0=\mathbf{0}$, $\varepsilon\le 1e-8$ and $t=0$.
2: **while** not converged **do**
3: Update $(\mathcal{L}^{t+1},\mathcal{S}^{t+1})$ by Eq. (34);
4: Update $\mathcal{K}^{t+1}$ by Eq. (35);
5: Update $\mathcal{T}^{t+1}$ by Eq. (37);
6: Update $\mathcal{M}^{t+1}$ by Eq. (40);
7: Update $\mathcal{Y}_1^{t+1},\mathcal{Y}_2^{t+1},\mathcal{Y}_3^{t+1}$ by Eq. (41);
8: Stop criterion: $\|\mathcal{K}^{t+1}-\mathcal{L}^{t+1}\|_{l_\infty}\le\varepsilon$, $\|\mathcal{T}^{t+1}-\mathcal{S}^{t+1}\|_{l_\infty}\le\varepsilon$, $\|\mathcal{M}^{t+1}-\mathcal{L}^{t+1}\|_{l_\infty}\le\varepsilon$, and $\max\left\{\|\mathcal{X}^{t+1}-\mathcal{X}^t\|_{l_\infty}\right\}\le\varepsilon,\forall\mathcal{X}\in\{\mathcal{L},\mathcal{S},\mathcal{K},\mathcal{T},\mathcal{M}\}$.
9: $t=t+1$.
10: **end while**

Computational complexity. In a single iteration, the main cost comes from updating $\mathcal{L}^t$ which involves computing FFT, IFFT and d_3 SVDs of $d_1\times d_2$ matrices [19]. Hence Algorithm 1 has per-iteration complexity of order $O\big(d_1d_2d_3(d_1\wedge d_2+\log d_3)\big)$. Thus, if the total iteration number is T, then the total computational complexity is

$$O\Big(Td_1d_2d_3(d_1\wedge d_2+\log d_3)\Big). \tag{42}$$

Convergence analysis. According to [31], the convergence rate of general ADMM-based algorithms is $O(1/t)$, where t is the iteration number. The convergence of Algorithm 1 is analyzed in the following theorem.

Theorem 4 (Convergence of Algorithm 1). *For any $\rho>0$, if the unaugmented Lagrangian $L_0(\mathcal{L},\mathcal{S},\mathcal{K},\mathcal{T},\mathcal{M},\mathcal{Y}_1,\mathcal{Y}_2,\mathcal{Y}_3)$ of Problem (32) has a saddle point, then the iteration $(\mathcal{L}^t,\mathcal{S}^t,\mathcal{K}^t,\mathcal{T}^t,\mathcal{M}^t,\mathcal{Y}_1^t,\mathcal{Y}_2^t,\mathcal{Y}_3^t)$ in Algorithm 1 satisfies the residual convergence, objective convergence and dual variable convergence of Problem (32)*[4].

5.2. A Frank-Wolfe-based algorithm

The one iteration cost of Algorithm 1 goes superlinearly with the tensor size, which may be expensive for large tensors. Motivated by [32], we propose using a modified Frank-Wolfe algorithm to reduce the one-iteration cost.

[2] By using the conjugate symmetry of DFT [23], Eq. (36) can be performed with $\lceil\frac{d_3+1}{2}\rceil$ (rather than d_3) full SVDs of $d_1\times d_2$ matrices in the Fourier domain (see Algorithm 3 in [6]).

[3] Note that we have defined $\frac{0}{0}=0$ if the denominator is 0. This also applies to the computation of $\mathcal{B}$ in Eq. (39).

[4] See the supplemental material for the detailed explanation of "residual convergence, objective convergence and dual variable convergence".

In Problem (13), the l_∞-norm constraint on $\mathcal{L}$ serves an incoherence condition. In real applications, one often omits this constraint and considers the following unconstrained problem:

$$(\hat{\mathcal{L}}, \hat{\mathcal{S}}) = \underset{\mathcal{L},\mathcal{S}}{\operatorname{argmin}} \frac{1}{2}\|\mathcal{L}+\mathcal{S}-\mathcal{Y}\|_{\mathrm{F}}^2 + \lambda\|\mathcal{L}\|_\star + \mu R(\mathcal{S}). \tag{43}$$

By introducing two upper bounds on $\|\mathcal{L}^*\|_\star$ and $R(\mathcal{S}^*)$: $u_l \geq \|\mathcal{L}^*\|_\star$ and $u_s \geq R(\mathcal{S}^*)$, and two intermediate variables t_l and t_s, Problem (43) is equivalent to the following problem:

$$\begin{aligned} \min_{\mathcal{L},\mathcal{S},t_l,t_s} \quad & F(\mathcal{L},\mathcal{S},t_l,t_s) := \tfrac{1}{2}\|\mathcal{Y}-\mathcal{L}-\mathcal{S}\|_{\mathrm{F}}^2 + \lambda t_l + \mu t_s \\ \text{s.t.} \quad & \|\mathcal{L}\|_\star \leq t_l \leq u_l, R(\mathcal{S}) \leq t_s \leq u_s. \end{aligned} \tag{44}$$

5.2.1. The Frank-Wolfe method

The Frank-Wolfe (FW) method [33,34], also known as the conditional gradient method, applies to the minimization of a smooth function $h(\cdot)$ over a compact, convex domain $\mathcal{D} \subset \mathbb{R}^n$:

$$\min_{\mathbf{x}} h(\mathbf{x}) \qquad \text{s.t.} \quad \mathbf{x} \in \mathcal{D}. \tag{45}$$

Here, ∇h is assumed to be L-Lipschitz: $\|\nabla h(\mathbf{x}) - \nabla h(\mathbf{y})\|_2 \leq L\|\mathbf{x}-\mathbf{y}\|_2$, $\forall \mathbf{x}, \mathbf{y} \in \mathcal{D}$. Let $D = \max_{\mathbf{x},\mathbf{y}\in\mathcal{D}} \|\mathbf{x}-\mathbf{y}\|_2$ denote the diameter of the feasible set $\mathcal{D}$.

In its simplest form, FW first linearizes the smooth object function $h(\mathbf{x})$ at $\mathbf{x}^t$ of iteration t,

$$h(\mathbf{v}) \approx h(\mathbf{x}^t) + \langle \nabla h(\mathbf{x}^t), \mathbf{v}-\mathbf{x}^t \rangle. \tag{46}$$

Then, one minimizes the linear surrogate to obtain

$$\mathbf{v}^t \in \underset{\mathbf{v}\in\mathcal{D}}{\operatorname{argmin}} \langle \nabla h(\mathbf{x}^t), \mathbf{v} \rangle, \tag{47}$$

after which we update $\mathbf{x}^{t+1}$ as some point in $\mathcal{D}$ such that

$$h(\mathbf{x}^{t+1}) \leq h(\mathbf{x}^t + \gamma(\mathbf{v}^t - \mathbf{x}^t)), \tag{48}$$

where $\gamma = \frac{2}{t+2}$.

5.2.2. Modified FW

Inspired by [32], we develop a modified FW algorithm (Algorithm 2.) for Problem (43). The proposed FW-based algorithm consists of three steps: an FW step, an exact line search step, and a proximal gradient step for $\mathcal{S}$. The exact line search step seeks a better γ instead of directly using $\gamma = 2/(t+2)$ in Eq. (48) for further acceleration. The proximal gradient step for $\mathcal{S}$ is applied to overcome the problem of slow convergence of $\mathcal{S}^t$ caused by using the "vanilla" FW.

Algorithm 2 Solve Problem (44) by modified Frank-Wolfe method.

Require: Observation $\mathcal{Y}$, parameters λ, μ, u_l, u_s, and ε.

1: Initialize $\mathcal{L}^0 = \mathcal{S}^0 = \mathbf{0}$, $t_l^0 = t_s^0 = 0$, $\varepsilon \leq 1e-8$ and $t = 0$.
2: **while** not converged **do**
3: Update $(\mathcal{V}_l^t, v_l^t)$ by Eq.(49);
4: Update $(\mathcal{V}_s^t, v_s^t)$ by Eq.(50);
5: Update $(\mathcal{L}^{t+\frac{1}{2}}, \mathcal{S}^{t+\frac{1}{2}}, t_l^{t+\frac{1}{2}}, t_s^{t+\frac{1}{2}})$ by Eq.(56);
6: Update $\mathcal{S}^{t+1}$ by Eq.(57);
7: Let $\mathcal{L}^{t+1} = \mathcal{L}^{t+\frac{1}{2}}, t_l^{t+1} = t_l^{t+\frac{1}{2}}$, and $t_s^{t+1} = R(\mathcal{S}^{t+1})$.
8: Stop criterion: $F(\mathcal{L}^{t+1}, \mathcal{S}^{t+1}, t_l^{t+1}, t_s^{t+1}) - F(\mathcal{L}^t, \mathcal{S}^t, t_l^t, t_s^t) \leq \varepsilon F(\mathcal{L}^t, \mathcal{S}^t, t_l^t, t_s^t)$.
9: $t = t+1$.
10: **end while**

An FW Step. Following the key step Eq. (47) of FW, we first update $\mathbf{v}^t = (\mathcal{V}_l^t, \mathcal{V}_s^t, v_l^t, v_s^t)$ by:

$$(\mathcal{V}_l^t, v_l^t) \in \underset{\|\mathcal{V}_l\|_\star \leq v_l \leq u_l}{\operatorname{argmin}} g_l(\mathcal{V}_l, v_l) := \langle \mathcal{E}^t, \mathcal{V}_l \rangle + \lambda v_l, \tag{49}$$

$$(\mathcal{V}_s^t, v_s^t) \in \underset{R(\mathcal{V}_s) \leq v_s \leq u_s}{\operatorname{argmin}} g_s(\mathcal{V}_s, v_s) := \langle \mathcal{E}^t, \mathcal{V}_s \rangle + \mu v_s, \tag{50}$$

where $\mathcal{E}^t = \mathcal{L}^t + \mathcal{S}^t - \mathcal{Y}$ plays the role of $\nabla h(\mathbf{x}^t)$ in Eq. (47). To solve Problems (49) and (50), we come up with the following two lemmas.

Lemma 5. *Let $k^* = \operatorname{argmin}_{k\leq d_3} \|\widetilde{\mathcal{E}}^{(k)}\|$. Let $\mathbf{u} \in \mathbb{R}^{d_1}$ and $\mathbf{v} \in \mathbb{R}^{d_2}$ be one pair of the left and right singular vectors of $\tilde{\mathbf{A}}^{(k^*)} := \widetilde{\mathcal{A}}(:,:,k^*)$ corresponding to the leading singular value. Let $\mathcal{G}_l = \operatorname{real}(\operatorname{ifft}_3(\mathcal{B}))$, where $\mathcal{B}(:,:,k^*) = \mathbf{u}\mathbf{v}^H$ and $\mathcal{B}(:,:,k) = \mathbf{0}, \forall k \neq k^*$. Then, one solution point of $(\mathcal{V}_l^t, v_l^t)$ of Problem (49) can be given as*

$$(\mathcal{V}_l^t, v_l^t) = \begin{cases} (-u_l d_3 \mathcal{G}_l, u_l), & \|\mathcal{G}^t\|_{\mathrm{sp}} > \lambda, \\ (\mathbf{0}, 0), & \|\mathcal{G}^t\|_{\mathrm{sp}} \leq \lambda, \end{cases} \tag{51}$$

and the optimal value of Problem (49) is $-u_l\big(\|\mathcal{E}^t\|_{\mathrm{sp}} - \lambda, 0\big)_+$.

Lemma 6. *The optimal value of Problem (50) is $-u_s\big(R^*(\mathcal{E}^t) - \mu\big)_+$ and one particular solution point $(\mathcal{V}_s^t, v_s^t)$ of Problem (50) can be given as*

$$(\mathcal{V}_s^t, v_s^t) = \begin{cases} (-u_s \mathcal{G}_s, u_s), & R^*(\mathcal{E}^t) > \mu, \\ (\mathbf{0}, 0), & R^*(\mathcal{E}^t) \leq \mu, \end{cases} \tag{52}$$

where the intermediate variable $\mathcal{G}_s$ is computed as follows

(I). *If $R(\cdot) = \|\cdot\|_{l_1}$, then*

$$\mathcal{G}_s = \operatorname{sign}(\mathcal{E}_{i^*j^*k^*}^t)\mathbf{e}_{i^*} \circ \mathbf{e}_{j^*} \circ \mathbf{e}_{k^*}, \tag{53}$$

where $(i^*, j^*, k^*) \in \operatorname{argmax}_{(i,j,k)} |\mathcal{E}_{ijk}^t|$;

(II). *If $R(\cdot) = \|\cdot\|_{\mathrm{tube}_1}$, then*[5]

$$\mathcal{G}_s(i,j,:) = \begin{cases} \frac{\mathcal{E}^t(i^*,j^*,:)}{\|\mathcal{E}^t(i^*,j^*,:)\|_2}, & \text{if } (i,j) = (i^*,j^*) \\ \mathbf{0}, & \text{otherwise.} \end{cases} \tag{54}$$

where $(i^*, j^*) \in \operatorname{argmax}_{(i,j)} \|\mathcal{E}^t(i,j,:)\|_2$.

(III). *If $R(\cdot) = \|\cdot\|_{\mathrm{slice}_1}$, then*

$$\mathcal{G}_s(:,j,:) = \begin{cases} \frac{\mathcal{E}^t(:,j^*,:)}{\|\mathcal{E}^t(:,j^*,:)\|_{\mathrm{F}}}, & \text{if } j = j^* \\ \mathbf{0}, & \text{otherwise.} \end{cases} \tag{55}$$

where $j^* \in \operatorname{argmax}_j \|\mathcal{E}^t(:,j,:)\|_{\mathrm{F}}$.

Exact line search. Using line search [32], we then update $(\mathcal{L}^{t+\frac{1}{2}}, \mathcal{S}^{t+\frac{1}{2}}, t_l^{t+\frac{1}{2}}, t_s^{t+\frac{1}{2}})$ by

$$\begin{aligned} \min_{\substack{\mathcal{L},\mathcal{S},t_l,t_s, \\ \gamma_1,\gamma_2\in[0,1]}} \quad & F(\mathcal{L},\mathcal{S},t_l,t_s) \\ \text{s.t.} \quad & \begin{pmatrix} \mathcal{L} \\ t_l \end{pmatrix} = (1-\gamma_1)\begin{pmatrix} \mathcal{L}^t \\ t_l^t \end{pmatrix} + \gamma_1 \begin{pmatrix} \mathcal{V}_l^t \\ v_l^t \end{pmatrix} \\ & \begin{pmatrix} \mathcal{S} \\ t_s \end{pmatrix} = (1-\gamma_2)\begin{pmatrix} \mathcal{S}^t \\ t_s^t \end{pmatrix} + \gamma_2 \begin{pmatrix} \mathcal{V}_s^t \\ v_s^t \end{pmatrix}. \end{aligned} \tag{56}$$

By solving the quadratic problem (56), we use the following γ_1 and γ_2 to further compute $(\mathcal{L}^{t+\frac{1}{2}}, \mathcal{S}^{t+\frac{1}{2}}, t_l^{t+\frac{1}{2}}, t_s^{t+\frac{1}{2}})$

$$\gamma_1 = \begin{cases} 0, & \tilde{\gamma}_1 < 0 \\ \tilde{\gamma}_1, & \tilde{\gamma}_1 \in [0,1] \\ 1, & \tilde{\gamma}_1 > 1 \end{cases}, \quad \gamma_2 = \begin{cases} 0, & \tilde{\gamma}_2 < 0 \\ \tilde{\gamma}_2, & \tilde{\gamma}_2 \in [0,1] \\ 1, & \tilde{\gamma}_2 > 1 \end{cases},$$

where if $\|\mathcal{A}\|_{\mathrm{F}}^2\|\mathcal{B}\|_{\mathrm{F}}^2 = \langle \mathcal{A}, \mathcal{B} \rangle^2$, we choose $\tilde{\gamma}_1 = \tilde{\gamma}_2 = t/(t+2)$; otherwise, we choose

$$\tilde{\gamma}_1 = \frac{\big(\langle \mathcal{B}, \mathcal{C} \rangle + e\big)\langle \mathcal{A}, \mathcal{B} \rangle - \big(\langle \mathcal{A}, \mathcal{C} \rangle + d\big)\|\mathcal{B}\|_{\mathrm{F}}^2}{\|\mathcal{A}\|_{\mathrm{F}}^2\|\mathcal{B}\|_{\mathrm{F}}^2 - \langle \mathcal{A}, \mathcal{B} \rangle^2},$$

$$\tilde{\gamma}_2 = \frac{\big(\langle \mathcal{B}, \mathcal{C} \rangle + e\big)\|\mathcal{A}\|_{\mathrm{F}}^2 - \big(\langle \mathcal{A}, \mathcal{C} \rangle + d\big)\langle \mathcal{A}, \mathcal{B} \rangle}{\langle \mathcal{A}, \mathcal{B} \rangle^2 - \|\mathcal{A}\|_{\mathrm{F}}^2\|\mathcal{B}\|_{\mathrm{F}}^2},$$

[5] Note that we have defined $\frac{0}{0} = 0$ if the denominator is 0. This also applies to the computation of $\mathcal{G}_s(:,j,:)$ in Eq. (55).

with $\mathcal{A}=\mathcal{V}_l^t-\mathcal{L}^t$, $\mathcal{B}=\mathcal{V}_s^t-\mathcal{S}^t$, $\mathcal{C}=\mathcal{L}^t+\mathcal{S}^t-\mathcal{Y}$, $d=\lambda(v_l^t-t_l^t)$, and $e=\mu(v_s^t-t_s^t)$.

Proximal gradient step for $\mathcal{S}$. To update $\mathcal{S}$ in a more efficient way, we incorporate an additional proximal gradient step for $\mathcal{S}$. At iteration t, let $(\mathcal{L}^{t+\frac{1}{2}},\mathcal{S}^{t+\frac{1}{2}})$ be the result produced by FW step. To produce the next iterate, we keep the low-rank term $\mathcal{L}^{t+\frac{1}{2}}$, but use an extra proximal gradient step for the function $f(\mathcal{L}^{t+\frac{1}{2}},\mathcal{S})$ at point $\mathcal{S}^{t+\frac{1}{2}}$ to update $\mathcal{S}$, that is

$$\begin{aligned}\mathcal{S}^{t+1} &\in \underset{\mathcal{S}}{\operatorname{argmin}}\ F(\mathcal{L}^{t+\frac{1}{2}},\mathcal{S},t_l^{t+\frac{1}{2}},R(\mathcal{S}))\\ &=\underset{\mathcal{S}}{\operatorname{argmin}}\langle\mathcal{L}^{t+\frac{1}{2}}+\mathcal{S}^{t+\frac{1}{2}}-\mathcal{Y},\mathcal{S}-\mathcal{S}^{t+\frac{1}{2}}\rangle\\ &\quad+\frac{1}{2}\|\mathcal{S}-\mathcal{S}^{t+\frac{1}{2}}\|_{\mathrm{F}}^2+\mu R(\mathcal{S})\\ &=\operatorname{Prox}_{\mu}^{R(\cdot)}\left(\mathcal{Y}-\mathcal{L}^{t+\frac{1}{2}}\right).\end{aligned}\tag{57}$$

The algorithm is summarized in Algorithm 2 and the computational complexity and convergence behavior are analyzed as follows.

Computational complexity. The main cost lies in solving the subproblem (49) in Lemma 5. Only FFT/IFFT and d_3 pairs of leading singular vectors are computed. By using the conjugate symmetry of DFT [23], subproblem (49) can also be solved with $\lceil\frac{d_3+1}{2}\rceil$ (rather than d_3) rank-1 SVDs of $d_1\times d_2$ matrices in the Fourier domain. Thus, the per-iteration cost of Algorithm 2 is

$$O\left(d_1d_2d_3\log d_3\right).\tag{58}$$

It is significantly lower than $O\left(d_1d_2d_3(\min\{d_1,d_2\}+\log d_3)\right)$ which is the per-iteration cost of Algorithm 1.

Theorem 5 (Convergence of Algorithm 2)**.** *Let $(\mathcal{L}^\star,\mathcal{S}^\star,t_l^\star,t_s^\star)$ be the optimal solution of Problem (44). Then the sequence $(\mathcal{L}^t,\mathcal{S}^t,t_l^t,t_s^t)$ produced by Algorithm 2 satisfies*

$$F(\mathcal{L}^t,\mathcal{S}^t,t_l^t,t_s^t)-F(\mathcal{L}^\star,\mathcal{S}^\star,t_l^\star,t_s^\star)\le\frac{20(d_3u_l^2+u_s^2)}{t+2}.\tag{59}$$

According to Theorem 5, the convergence rate of Algorithm 2 is approximately $O(1/t)$, which is of the same order as our ADMM-based Algorithm 1. Considering the much lower per-iteration cost of Algorithm 2 than Algorithm 1, we may expect that Algorithm 2 can run mush faster than Algorithm 1. This expectation will be confirmed through experiments in Section 6.2.

6. Experiments

In this section, the correctness of the proposed error bounds in Theorem 2 is first verified through simulation studies. The effectiveness and efficiency of the proposed algorithms (i.e., Algorithms 1 and 2) are then evaluated through extensive experiments on real datasets. All codes are written in MATLAB and all experiments are performed in Windows 10 based on Intel(R) Core(TM) i7-8565U 1.80-1.99 GHz CPU with 16G RAM.

6.1. Correctness of the proposed error bounds

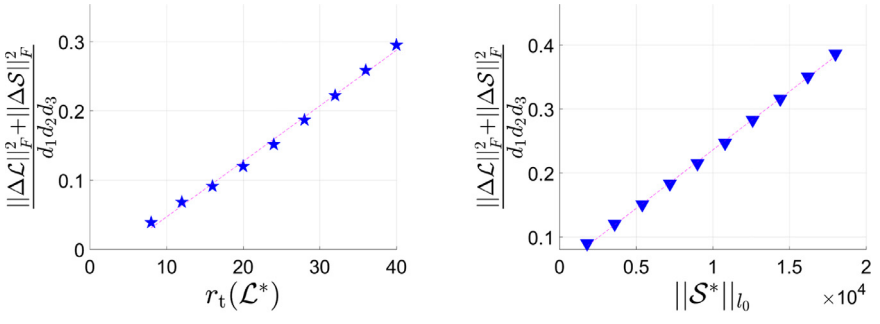
To validate the correctness of the upper bounds in Eqs. (23)--(25), we conduct simulations to check whether the proposed upper bounds can predict the right scaling behavior of the estimation errors.

Given the tubal rank $r^*\le d_1\wedge d_2$, the true tensor $\mathcal{L}^*\in\mathbb{R}^{d_1\times d_2\times d_3}$ is first formed by $\mathcal{L}^*=\mathcal{P}*\mathcal{Q}/d_3$, where the elements of tensors $\mathcal{P}\in\mathbb{R}^{d_1\times r^*\times d_3}$ and $\mathcal{Q}\in\mathbb{R}^{r^*\times d_2\times d_3}$ are sampled from *i.i.d.* standard Gaussian distribution. Then, we generate the outlier tensor $\mathcal{S}^*$ by choosing its support uniformly at random when $\mathcal{S}^*$ is element-wisely sparse. Similarly, we uniformly choose the tube or slice support at random for tube-wisely or slice-wisely sparse $\mathcal{S}^*$. The non-zero elements of $\mathcal{S}^*$ are sampled *i.i.d.* from a certain distribution. Further, we generate the noise tensor $\mathcal{E}$ with entries drawing *i.i.d.* from $\mathcal{N}(0,\sigma^2)$ with $\sigma=c\|\mathcal{L}^*\|_{\mathrm{F}}/\sqrt{d_1d_2d_3}$ to keep a constant signal noise ratio. Finally, we obtain the observation $\mathcal{Y}=\mathcal{L}^*+\mathcal{S}^*+\mathcal{E}$ according to the observation model (11).

For simplicity, we consider f-square tensors, i.e., $d_1=d_2=d$. We test tensors of 12 different size by choosing $d_2\in\{60,\ 80,\ 100\}$ and $d_3=20$. We choose $r^*\in\{8,\ 12,\ 16,\ \cdots,\ 40\}$ to generate $\mathcal{L}^*$. We generate the outlier tensor with sparsity ratio $\boldsymbol{\varrho}_{\mathrm{s}}(\mathcal{S}^*)\in\{0.025,0.05,\cdots,0.25\}$. We consider three different settings where the non-zero elements of $\mathcal{S}^*$ are drawn *i.i.d.* from $\mathrm{Bin}(-1,+1)$, or $\mathcal{N}(0,1)$, or $\mathcal{U}[0,1]$. For the noise tensor $\mathcal{E}$, we set the signal noise ratio $c=0.1$. The parameter α in Problem (13) is simply set to its oracle value in each simulation. We test 10 times for each setting by running Algorithm 1 and computing the averaged estimation error.

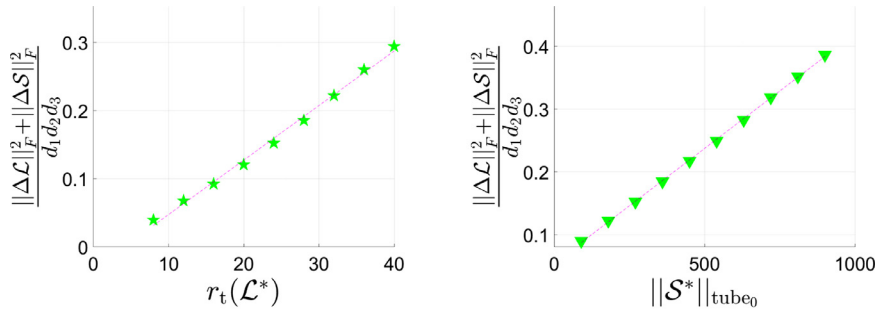
For tensors of a given size, it is predicted by Theorem 2 that upper bounds on the element-wise estimation error $\frac{\|\Delta_{\mathcal{L}}\|_{\mathrm{F}}^2+\|\Delta_{\mathcal{S}}\|_{\mathrm{F}}^2}{d_1d_2d_3}$ would scale approximately like $a_1r_{\mathrm{t}}(\mathcal{L}^*)+b_1\|\mathcal{S}^*\|_{l_0}$ for element-wisely sparse $\mathcal{S}^*$, $a_2r_{\mathrm{t}}(\mathcal{L}^*)+b_2\|\mathcal{S}^*\|_{\mathrm{tube}_0}$ for tube-wisely sparse $\mathcal{S}^*$, or $a_3r_{\mathrm{t}}(\mathcal{L}^*)+b_3\|\mathcal{S}^*\|_{\mathrm{slice}_0}$ for slice-wisely sparse $\mathcal{S}^*$, where $a_i,b_i,i=1,2,3$, are positive constants. Then, if the bounds are sharp, the real estimation errors would have the same scaling behavior. We will check whether these phenomena occur.

For tensors of size $60\times60\times20$, Fig. 4 shows the results of averaged element-wise estimation error versus tubal rank of $\mathcal{L}^*$ and tensor l_0-norm of $\mathcal{S}^*$, when $\mathcal{S}^*$ is element-wise with *i.i.d.* $\mathrm{Bin}(-1,+1)$ elements. We can see that the error has approximately linear scaling behavior with respect to $r_{\mathrm{t}}(\mathcal{L}^*)$ and $\|\mathcal{S}^*\|_{l_0}$.



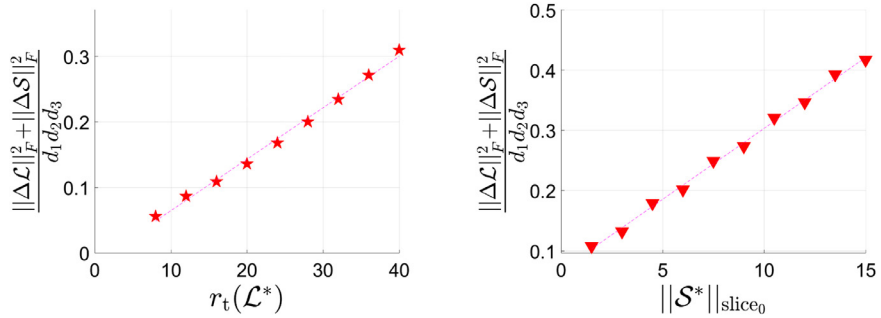
(a) (b)

Fig. 4. The averaged element-wise estimation error versus tubal rank of $\mathcal{L}^*$ and tensor l_0-norm of $\mathcal{S}^*$ for tensors of size $60\times60\times20$, when $\mathcal{S}^*$ is *element-wisely sparse* with *i.i.d.* $\mathrm{Bin}(-1,+1)$ elements. (a): Error vs $\mathcal{L}^*$, when $\|\mathcal{S}^*\|_{l_0}=1080$. (b): Error vs $\|\mathcal{S}^*\|_{l_0}$, when $r_{\mathrm{t}}(\mathcal{L}^*)=8$.



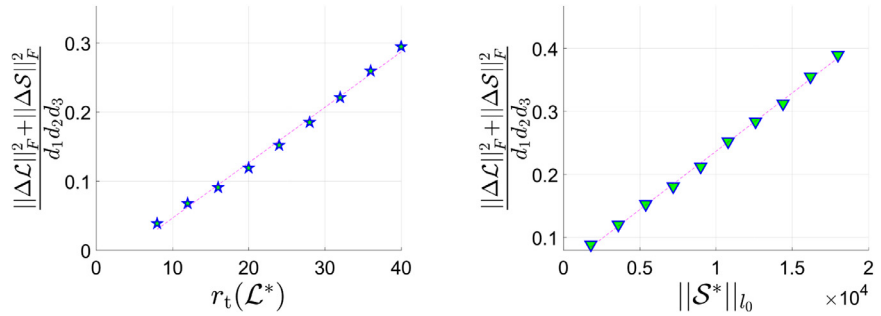
(a) (b)

Fig. 5. The averaged element-wise estimation error versus tubal rank of $\mathcal{L}^*$ and the number of non-zero tubes of $\mathcal{S}^*$ for tensors of size $60\times60\times20$, when $\mathcal{S}^*$ is *tube-wisely sparse* with *i.i.d.* Bin$(-1,+1)$ elements. (a): Error vs $\mathcal{L}^*$, when $\|\mathcal{S}^*\|_{\text{tube}_0}=360$. (b): Error vs $\|\mathcal{S}^*\|_{\text{tube}_0}$, when $r_t(\mathcal{L}^*)=8$.



(a) (b)

Fig. 6. The averaged element-wise estimation error versus tubal rank of $\mathcal{L}^*$ and the number of non-zero slices of $\mathcal{S}^*$ for tensors of size $60\times60\times20$, when $\mathcal{S}^*$ is *slice-wisely sparse* with *i.i.d.* Bin$(-1,+1)$ elements. (a): Error vs $\mathcal{L}^*$, when $\|\mathcal{S}^*\|_{\text{slice}_0}=1$. (b): Error vs $\|\mathcal{S}^*\|_{\text{slice}_0}$, when $r_t(\mathcal{L}^*)=8$.



(a) (b)

Fig. 7. The averaged element-wise estimation error versus tubal rank of $\mathcal{L}^*$ and tensor l_0-norm of $\mathcal{S}^*$ for tensors of size $60\times60\times20$, when $\mathcal{S}^*$ is element-wise with *i.i.d.* $\mathcal{N}(0,1)$ elements. (a): Error vs $\mathcal{L}^*$, when $\|\mathcal{S}^*\|_{l_0}=1080$. (b): Error vs $\|\mathcal{S}^*\|_{l_0}$, when $r_t(\mathcal{L}^*)=8$.

Thus, it can be said that the experimental results are consistent with our expectation for $d=60$. Figs. 5 and 6 show respectively the results when $\mathcal{S}^*$ is tube-wisely or slice-wisely sparse, and similar linear scaling behaviors are observed. For tensors of size $60\times60\times20$, Figs. 7 and 8 show the results for element-wisely sparse $\mathcal{S}^*$ with *i.i.d.* $\mathcal{N}(0,1)$ and $\mathcal{U}[0,1]$ elements, receptively. We can find that the linear scaling behavior also holds for different outlier distributions. When $\mathcal{S}^*$ is element-wise with *i.i.d.* Bin$(-1,+1)$ elements, Fig. 9 shows the results for tensors of size $100\times100\times20$, and the error also scales linearly with $r_t(\mathcal{L}^*)$ and $\|\mathcal{S}^*\|_{l_0}$. Similar phenomena have be found in other settings and we omit them due to space limitation. Thus it can be verified that the proposed bounds can approximately predict the scaling behavior of the estimation error.

6.2. Effectiveness and efficiency of the proposed algorithms

To show the superiority of Algorithms 1 and 2 for the proposed TNN-based RTD model (13), we conduct robust tensor recovery experiments on color images, point cloud data, and videos. We also compare with the tensor Schatten-1 norm based RTD model (SNN[6]) [5], and the matrix nuclear norm (NN) based robust matrix decomposition model [29] in both accuracy and running time. Since the source code of SNN [5] and NN [29] is not available, we formulate the corresponding models by using the aforementioned

[6] Different from [5], we consider a weighted version of SNN in which the nuclear norm the unfolding matrix along each mode is weighted by a positive vector $\boldsymbol{\alpha}$ satisfying $\sum_i\alpha_i=1$ [35].

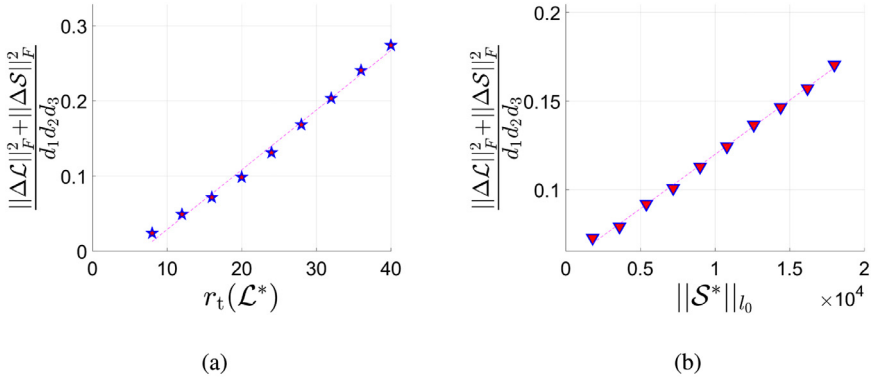


Fig. 8. The averaged element-wise estimation error versus tubal rank of $\mathcal{L}^*$ and tensor l_0-norm of $\mathcal{S}^*$ for tensors of size $60\times 60\times 20$, when $\mathcal{S}^*$ is element-wise with *i.i.d.* $\mathcal{U}[0,1]$ elements. (a): Error vs $\mathcal{L}^*$, when $\|\mathcal{S}^*\|_{l_0}=1080$. (b): Error vs $\|\mathcal{S}^*\|_{l_0}$, when $r_t(\mathcal{L}^*)=8$.

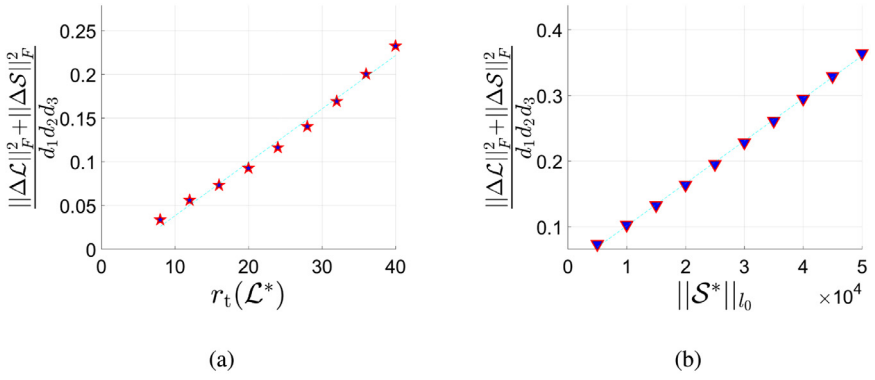


Fig. 9. The averaged element-wise estimation error versus tubal rank of $\mathcal{L}^*$ and tensor l_0-norm of $\mathcal{S}^*$ for tensors of size $100\times 100\times 20$, when $\mathcal{S}^*$ is element-wise with *i.i.d.* $\mathrm{Bin}(-1,+1)$ elements. (a): Error vs $\mathcal{L}^*$, when $\|\mathcal{S}^*\|_{l_0}=5000$. (b): Error vs $\|\mathcal{S}^*\|_{l_0}$, when $r_t(\mathcal{L}^*)=8$.

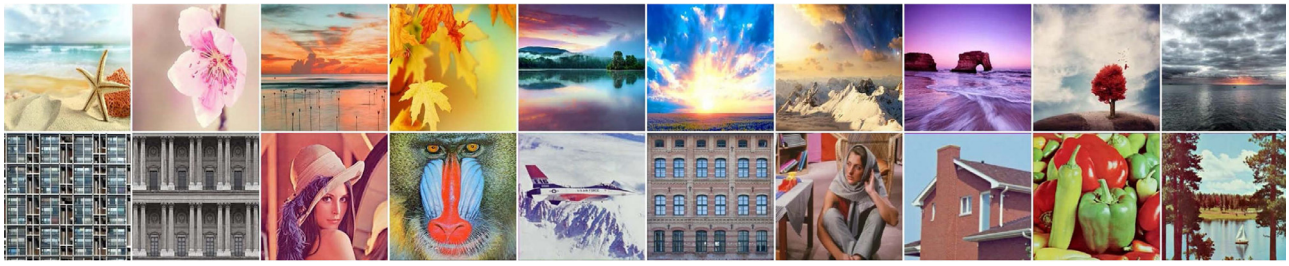


Fig. 10. Twenty test images.

norms to replace TNN in Problem (13), and solve the relevant optimization problems within the ADMM framework [36] through our implementations in MATLAB. To measure the quality of an estimated tensor $\hat{\mathcal{L}}$, the Peak Signal Noise Ratio (PSNR) defined as

$$\mathrm{PSNR} := 10\log_{10}\left(\frac{d_1d_2d_3\|\mathcal{L}^*\|_{l_\infty}^2}{\|\hat{\mathcal{L}}-\mathcal{L}^*\|_{\mathrm{F}}^2}\right)$$

is applied. Higher PSNR value means better estimation performance.

6.2.1. Color image recovery

In this experiment, we conduct robust tensor recovery on twenty color images of size $512\times 512\times 3$ (see Fig. 10). Three different settings of outliers, i.e., element-wise, tube-wise, or column-wise outliers, are considered. Specifically, for an image $\mathcal{L}^*\in\mathbb{R}^{m\times k\times 3}$, we first generate the outlier tensor $\mathcal{S}^*$ by choosing 10% of the support (or 10% of the tube-support, or 5% of the column-support) uniformly at random, and then corrupt the chosen elements by additive independent $\mathrm{Bin}(-1,+1)$ outliers. Then, we add noise tensor $\mathcal{E}$ of independent zero-mean Gaussian entries with standard deviation $\sigma=c\sigma_0$, where noise level $c=0.1$ or 0.2 and normalized signal magnitude $\sigma_0=\|\mathcal{L}^*\|_{\mathrm{F}}/\sqrt{3mk}$. Thus, the corrupted observation $\mathcal{Y}=\mathcal{L}^*+\mathcal{S}^*+\mathcal{E}$ are generated according to the observation model (11).

NN directly works on matrices of size 512×512, whereas SNN, TNN and FW works on tensors of size $512\times 512\times 3$. The parameter tuning is not an easy task, and the key parameters are set as follows. For NN, we set the regularization parameters $(\lambda,\mu)=(0.5, 0.5/\sqrt{\max\{m,k\}})$ for element-wise and tube-wise outliers (suggested by [2]), and $(\lambda,\mu)=(0.5, 0.5/\sqrt{\log(mk)})$ for column-wise outliers (suggested by [37]), respectively. For SNN, the weight parameters $\boldsymbol{\alpha}$ are chosen to satisfy $\alpha_1:\alpha_2:\alpha_3=1:1:c_1$ and $\sum\alpha_i=1$, where parameter c_1 is tuned in $\{0.01,\ 0.1,\ 0.3,\ 0.5\}$ for better performances in most cases; we set the regularization parameters $(\lambda,\mu)=(1,1/\sqrt{3m})$ for element-wise outliers, $(\lambda,\mu)=(1,1/\log(3mk))$ for tube-wise outliers, and $(\lambda,\mu)=(1,1/\log(3))$ for column-wise outliers, respectively. For TNN and FW, we respectively set the regularization

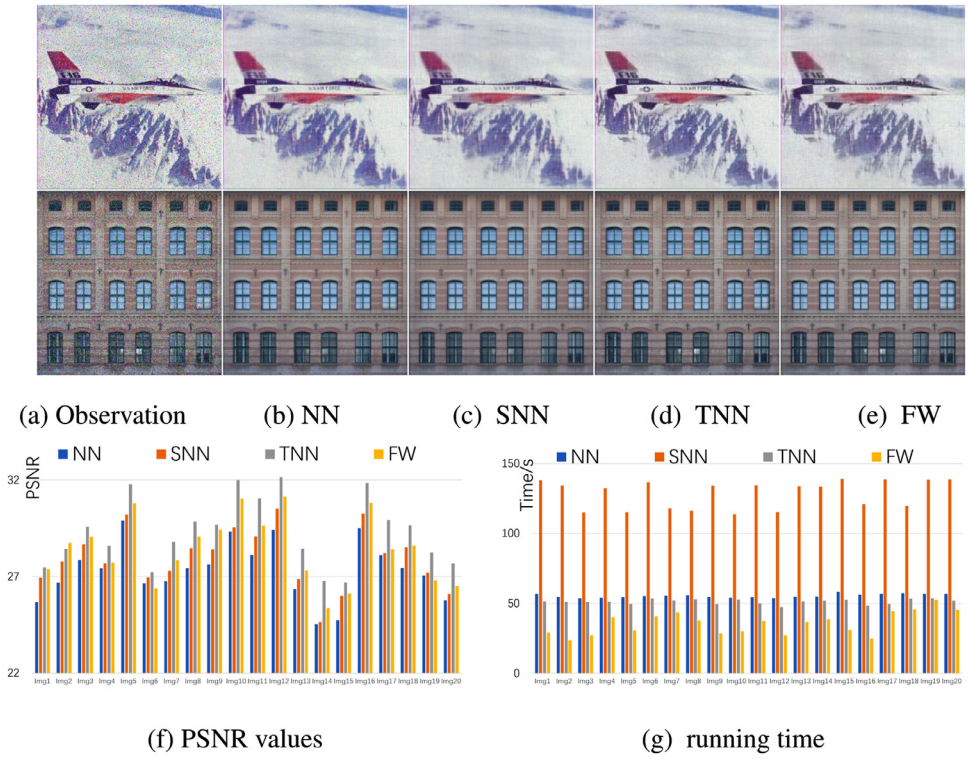


Fig. 11. Results of color image recovery with 10% of the *elements* corrupted by Bin$(-1,+1)$ outliers and all the elements polluted by Gaussian noise of level $c = 0.1$. (a) is the corrupted image; (b)-(e) are images recovered by NN [29], SNN [5] and the proposed TNN (Algorithm 1) and FW (Algorithm 2); (f) and (g) report the PSNR values and running time (seconds), respectively. **Best viewed in** 400% **zoomed color pdf file.**.

parameters $(\lambda, \mu) = (c_2, c_2/\sqrt{3\max\{m,k\}})$ for element-wise outliers (suggested by [19]), $(\lambda, \mu) = (c_2, c_2/\sqrt{\max\{m,k\}})$ for tube-wise outliers (motivated by [22]), and $(\lambda, \mu) = (c_2, 1.3c_2/\sqrt{\log k})$ for column-wise outliers (motivated by [8]); parameter c_2 is tuned in $\{2, 4, 0.008\|\mathcal{Y}\|_F\}$ (motivated by [32]) for better performances in most cases. For FW, we simply set parameters u_t and u_s as their oracle values, which can be reasonably considered as the (near)-optimal setting. The initializations and stop criteria of the algorithms are chosen to get a reasonably good performance/time balance. Given a color image and a corruption level, we test 10 times and report the averaged PSNR and time.

Both qualitative and quantitative results are shown in Figs. 11, 12, and 13 for element-wise, tube-wise, and column-wise outliers, respectively. It can be seen that the TNN has the highest PSNR values and FW runs the fastest. The experimental results are easy to interpret, and in consistence with image recovery experiments in [6]. Firstly, NN cannot exploit the inter-channel correlations, so it performs worse than the tensor models. Secondly, TNN outperforms SNN, which can be interpreted by the discussion in [6] that TNN adopts the low-tubal-rank assumption (or more precisely, low-average-rank assumption) which is weaker than the low-Tucker-rank assumption adopted by SNN. Thirdly, TNN can be faster than SNN in many circumstances because NN needs to compute full SVDs on three matrices of size $m \times k$, however TNN can only computes two full SVDs in the Fourier domain due to the conjugate symmetry of DFT (see Algorithm 3 in [6]). Finally, FW runs faster than TNN since it only involves computing the leading singular vectors instead of the full SVD in the Fourier domain.

6.2.2. Point cloud data set.

Point cloud data collected by light detection and ranging (LiDAR) sensors are widely used in environmental sensing for unmanned ground vehicles (UGV). In this experiment, we test on a dataset[7] for moving object tracking. It contains a sequence of point cloud data acquired from a Velodyne HDL-64E LiDAR. We choose the first 30 frames and form two tensors of size $64 \times 870 \times 30$ representing the distance data and the intensity data, respectively. We conduct robust tensor recovery against element-wisely sparse corruptions and Gaussian noise. Specifically, we first generate the outlier tensor $\mathcal{S}^*$ by choosing $\boldsymbol{\varrho}_s \in \{10\%, 20\%, 30\%\}$ of the entries of a zero tensor $\underline{\boldsymbol{0}}$ uniformly at random, and then fill in the chosen positions by independent Bin$(-1,+1)$ outliers. Then, we form the noise tensor $\mathcal{E}$ of independent zero-mean Gaussian entries with standard deviation $\sigma = c\sigma_0$, where noise level $c = 0.1$ and normalized signal magnitude $\sigma_0 = \|\mathcal{L}^*\|_F/\sqrt{d_1 d_2 d_3}$ for tensor signal $\mathcal{L}^* \in \mathbb{R}^{d_1 \times d_2 \times d_3}$. Thus, the corrupted observation $\mathcal{Y} = \mathcal{L}^* + \mathcal{S}^* + \mathcal{E}$ are generated according to the observation model (11).

NN directly works on frontal slices, whereas SNN, TNN and FW works on tensors. The key parameters are set as follows. For NN, we set the regularization parameters $(\lambda, \mu) = (0.5, 0.5/\sqrt{\max\{d_1, d_2\}})$ (suggested by [2]). For SNN, the weight parameters $\boldsymbol{\alpha}$ are tuned to satisfy $\alpha_1 : \alpha_2 : \alpha_3 = 1 : 1 : 0.3$, and we tune the regularization parameters $(\lambda, \mu) = (2, 5.2/\sqrt{\max\{d_1, d_2\}d_3})$ for better performances in most cases. For TNN and FW, we set the regularization parameters $(\lambda, \mu) = (c', c'/\sqrt{\max\{d_1, d_2\}d_3})$, where parameter c' is tuned in $\{2, 0.05\|\mathcal{Y}\|_F\}$ (motivated by [32]) for better performances in most cases. For FW, we simply set parameters u_t and u_s as their oracle values. The initializations and stop criteria of the algorithms are set for a reasonably good performance/time balance. We repeat 10 runs in each setting and report the averaged PSNR and time in Table 2. We can see that TNN has the highest PSNR values and FW

[7] *Scenario B* and *Scenario B-additional* dataset from http://www.mrt.kit.edu/z/publ/download/velodynetracking/dataset.html.

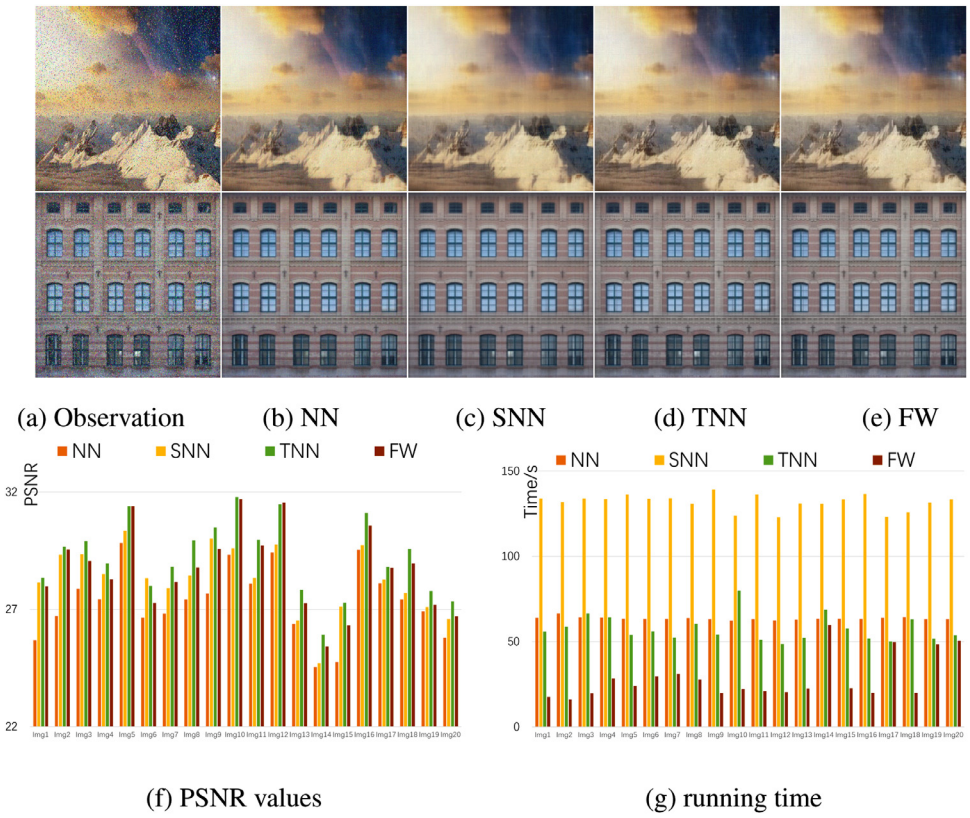


Fig. 12. Results of color image recovery with 10% of the *tubes* corrupted by Bin$(-1,+1)$ outliers and all the elements polluted by Gaussian noise of level $c = 0.1$. (a) is the corrupted image; (b)-(e) are images recovered by NN [29], SNN [5] and the proposed TNN (Algorithm 1) and FW (Algorithm 2); (f) and (g) report the PSNR values and running time (seconds) of the test images, respectively. **Best viewed in** 400% **zoomed color pdf file.**.

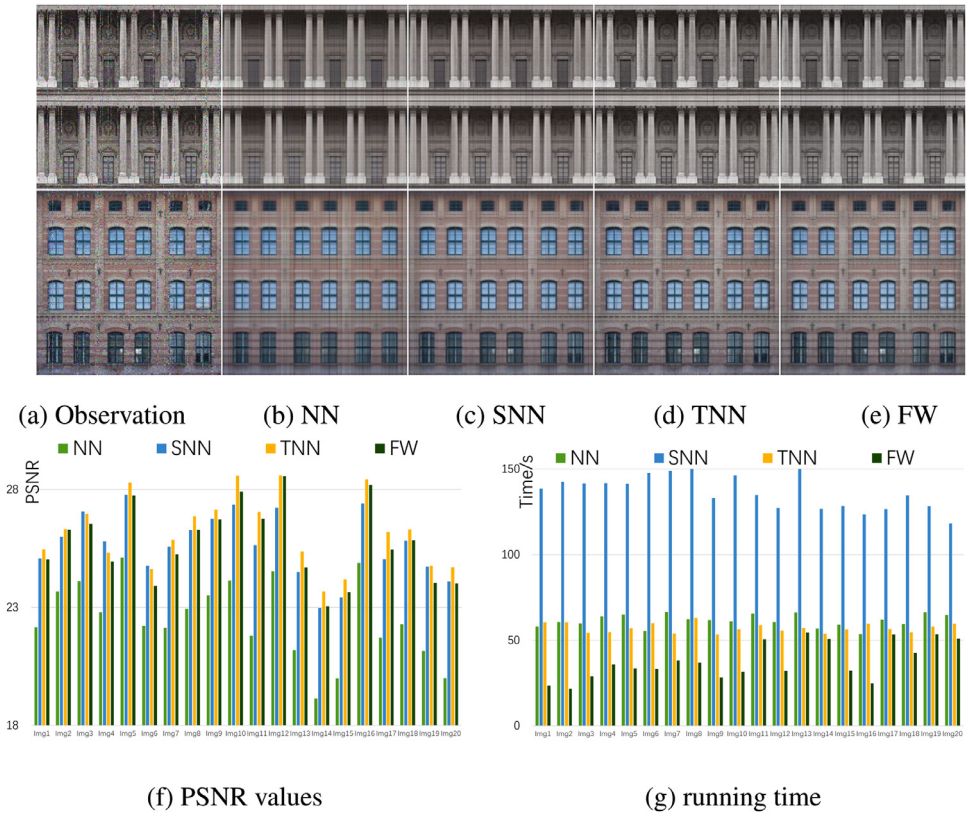


Fig. 13. Results of color image recovery with 5% of the *literal slices (i.e. columns)* corrupted by Bin$(-1,+1)$ outliers and all the elements polluted by Gaussian noise of level $c = 0.2$. (a) is the corrupted image; (b)-(e) are images recovered by NN [29], SNN [5] and the proposed TNN (Algorithm 1) and FW (Algorithm 2); (f) and (g) report the PSNR values and running time (seconds), respectively. **Best viewed in** 400% **zoomed color pdf file.**.

Table 2
PSNR values and running time (seconds) of different algorithms on point cloud data. The signal tensor is first corrupted by element-wise Bin$(-1,+1)$ outliers with corruption ratio $\varrho_s \in \{10\%, 20\%, 30\%\}$, and then all the elements are polluted by Gaussian noise with noise level $c = 0.1$.

Data set	cor. ratio ϱ_s	index	NN [29]	SNN [5]	TNN (Algorithm 1)	FW (Algorithm 2)
HDL Distance	10%	PSNR	20.25	25.41	**26.24**	25.91
		time/s	50.25	132.14	56.86	16.25
	20%	PSNR	19.77	22.23	**26.07**	25.35
		time/s	54.83	125.82	56.71	13.97
	30%	PSNR	19.3	21.45	**25.68**	24.67
		time/s	50.46	127.53	54.27	12.78
HDL Intensity	10%	PSNR	18.44	22.49	**23.13**	22.97
		time/s	56.11	135.22	64.23	22.14
	20%	PSNR	18.09	20.15	**22.88**	22.45
		time/s	55.90	126.85	62.12	18.65
	30%	PSNR	17.71	19.37	**22.54**	21.71
		time/s	57.29	128.38	58.47	19.52

Table 3
PSNR values and running time (seconds) of different algorithms on video data. First 10% entries of the video is corrupted by element-wise Bin$(-1,+1)$ outliers and then all the entries are polluted by Gaussian noise with noise level $c = \{0.1, 0.2\}$.

Data set	noise. level c	index	NN [29]	SNN [5]	TNN (Algorithm 1)	FW (Algorithm 2)
Claire	0.1	PSNR	25.68	28.67	**30.26**	29.70
		time/s	52.70	71.28	63.19	20.04
	0.2	PSNR	24.26	27.33	**30.06**	28.90
		time/s	51.06	71.88	68.61	21.91
Grandma	0.1	PSNR	23.21	29.82	**31.18**	29.24
		time/s	48.89	70.62	33.16	22.84
	0.2	PSNR	22.98	28.85	**31.05**	28.52
		time/s	51.39	69.09	37.24	20.68
Miss-America	0.1	PSNR	25.59	29.31	**31.36**	31.34
		time/s	53.70	68.07	33.18	19.74
	0.2	PSNR	25.33	28.64	**31.28**	30.86
		time/s	61.89	84.70	45.61	28.84

runs the fastest, which is in consistence with the experiments on color images.

6.2.3. Video recovery

In this subsection, we conduct video restoration which aims to recover an underlying video from its corrupted observation. The experiments are carried out on three widely used YUV videos[8]: Claire_qcif, Grandma_qcif, and Miss-America_qcif. We use the first 30 frames of Y components in each video and obtain three tensors sized $144 \times 176 \times 30$. We first choose 10% of video entries randomly, and corrupt them by additive independent Bin$(-1,+1)$ outliers. Then, we add *i.i.d.* zero-mean Gaussian noise with standard deviation $\sigma = c\sigma_0$, where the noise level $c \in \{0.1, 0.2\}$. The key parameters are set in a similar manner as the experiments on point cloud data. We report the averaged PSNR and time over 10 runs in Table 3. It can be found that the TNN has the highest PSNR values and FW runs the fastest, illustrating the effectiveness and efficiency of the proposed algorithms.

7. Conclusion

Inspired by the superior performance of low tubal rank tensor models, a TNN-based estimator is first proposed to recover a signal tensor against both small noises and sparse outliers. Statistically, we establish both deterministic and non-asymptotic upper bounds on the estimation error. We further show the non-asymptotic upper bounds are minimax near-optimal. Computationally, we develop an ADMM-based algorithm and an FW-based algorithm to solve the model with convergence guarantees. The FW-based algorithm takes the advantages of the dual norm of TNN which get rids of computing full SVDs in each iteration, and thus accelerates the algorithm. Experimentally, simulations on synthetic dataset verify the correctness of the theorem. The effectiveness and the efficiency of the proposed algorithms are evaluated on real datasets. An interesting future direction is to consider tensor estimation in the saturation setting. Another direction is to combine tensor learning with discrimination [38] and structured sparse representation [39] for face recognition.

Declaration of Competing Interest

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

Acknowledgments

The authors are grateful to the anonymous reviewers for their insightful comments and suggestions that highly improve the quality of this paper. The authors would like to thank Ms. Min Niu, Ms. Jin Wang, Mr. Bo Wang, and Mr. Dongxu Wei for their long-time accompany and support. This work is partially supported by the National Natural Science Foundation of China [Grant Nos. 61872188, U1713208, 61602244, 61672287, 61702262, 61773215, 61703209].

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.sigpro.2019.107319.

[8] Available from https://sites.google.com/site/subudhibadri/fewhelpfuldownloads.

References

[1] A. Cichocki, D. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, H.A. Phan, Tensor decompositions for signal processing applications: from two-way to multiway component analysis, IEEE SPM 32 (2) (2015) 145–163.

[2] E.J. Candés, X. Li, Y. Ma, J. Wright, Robust principal component analysis? JACM 58 (3) (2011) 11.

[3] L.R. Tucker, Some mathematical notes on three-mode factor analysis, Psychometrika 31 (3) (1966) 279–311.

[4] Q. Zhao, D. Meng, X. Kong, Q. Xie, W. Cao, Y. Wang, Z. Xu, A novel sparsity measure for tensor recovery, in: ICCV, 2015, pp. 271–279.

[5] Q. Gu, H. Gui, J. Han, Robust tensor decomposition with gross corruption, in: NIPS, 2014, pp. 1422–1430.

[6] C. Lu, J. Feng, W. Liu, Z. Lin, S. Yan, et al., Tensor robust principal component analysis with a new tensor nuclear norm, IEEE TPAMI (2019).

[7] J.Q. Jiang, M.K. Ng, Exact tensor completion from sparsely corrupted observations via convex optimization, 2017 arXiv:1708.00601.

[8] P. Zhou, J. Feng, Outlier-robust tensor PCA, CVPR, 2017.

[9] D. Goldfarb, Z. Qin, Robust low-rank tensor recovery: models and algorithms, SIAM J. Matrix Anal. Appl. 35 (1) (2014) 225–253.

[10] C.J. Hillar, L. Lim, Most tensor problems are np-hard, J. ACM 60 (6) (2009) 45.

[11] S. Friedland, L. Lim, Nuclear norm of higher-order tensors, Math. Comput. 87 (311) (2017) 1255–1281.

[12] R. Tomioka, T. Suzuki, K. Hayashi, H. Kashima, Statistical performance of convex tensor decomposition, in: NIPS, 2011, pp. 972–980.

[13] Z. Zhang, S. Aeron, Exact tensor completion using t-SVD, IEEE TSP 65 (6) (2017) 1511–1526.

[14] Z. Long, Y. Liu, L. Chen, C. Zhu, Low rank tensor completion for multiway visual data, Signal Process. 155 (2019) 301–316.

[15] W. Sun, L. Huang, H. So, J. Wang, Orthogonal tubal rank-1 tensor pursuit for tensor completion, Signal Process. 157 (2019) 213–224.

[16] W. Sun, Y. Chen, L. Huang, et al., Tensor completion via generalized tensor tubal rank minimization using general unfolding, IEEE Signal Process. Lett. (2018). 1–1

[17] C. Lu, J. Feng, Z. Lin, et al., Exact low tubal rank tensor recovery from gaussian measurements, in: IJCAI, 2018, pp. 1948–1954.

[18] A. Wang, X. Song, X. Wu, Z. Lai, Z. Jin, Generalized dantzig selector for low–tubal-rank tensor recovery, in: ICASSP, IEEE, 2019, pp. 3427–3431.

[19] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, S. Yan, Tensor robust principal component analysis: exact recovery of corrupted low-rank tensors via convex optimization, in: CVPR, 2016, pp. 5249–5257.

[20] O. Semerci, N. Hao, M.E. Kilmer, E.L. Miller, Tensor-based formulation and nuclear norm regularization for multienergy computed tomography, IEEE TIP 23 (4) (2014) 1678–1693.

[21] X.-Y. Liu, S. Aeron, V. Aggarwal, et al., Low-tubal-rank tensor completion using alternating minimization, 2016 arXiv:1610.01690.

[22] Z. Zhang, G. Ely, S. Aeron, N. Hao, M. Kilmer, Novel methods for multilinear data completion and de-noising based on tensor-SVD, in: CVPR, 2014, pp. 3842–3849.

[23] M.E. Kilmer, K. Braman, N. Hao, R.C. Hoover, Third-order tensors as operators on matrices: a theoretical and computational framework with applications in imaging, SIAM J. Matrix Anal. Appl. 34 (1) (2013) 148–172.

[24] A. Wang, Z. Lai, Z. Jin, Noisy low-tubal-rank tensor completion, Neurocomputing 330 (2019) 267–279.

[25] A. Wang, D. Wei, B. Wang, Z. Jin, Noisy low-tubal-rank tensor completion through iterative singular tube thresholding, IEEE Access 6 (2018) 35112–35128.

[26] Z. Zhang, D. Liu, S. Aeron, A. Vetro, An online tensor robust PCA algorithm for sequential 2d data, in: ICASSP, 2016, pp. 2434–2438.

[27] O. Klopp, Noisy low-rank matrix completion with general sampling distribution, Bernoulli 20 (1) (2014) 282–303.

[28] O. Klopp, K. Lounici, A. Tsybakov, Robust matrix completion, Probab. Theory Related Field. (2017) 1–42.

[29] A. Agarwal, S. Negahban, M.J. Wainwright, Noisy matrix decomposition via convex relaxation: optimal rates in high dimensions, Ann. Stat. (2012) 1171–1197.

[30] A. Wang, Z. Jin, Near-optimal noisy low-tubal-rank tensor completion via singular tube thresholding, in: ICDM workshop, 2017, pp. 553–560.

[31] B. He, X. Yuan, On the o(1/n) convergence rate of the douglas–rachford alternating direction method, SIAM J. Numer. Anal. 50 (2) (2012) 700–709.

[32] C. Mu, Y. Zhang, J. Wright, D. Goldfarb, Scalable robust matrix recovery: Frank-Wolfe meets proximal methods, SIAM J. Sci. Comput. 38 (5) (2016) A3291–A3317.

[33] M. Jaggi, Revisiting Frank-Wolfe: projection-free sparse convex optimization., in: ICML, 2013, pp. 427–435.

[34] M. Frank, P. Wolfe, An algorithm for quadratic programming, Naval Res. Logist. Q. 3 (1–2) (1956) 95–110.

[35] J. Liu, P. Musialski, P. Wonka, J. Ye, Tensor completion for estimating missing values in visual data, IEEE TPAMI 35 (1) (2013) 208–220.

[36] S. Boyd, N. Parikh, et al., Distributed optimization and statistical learning via the alternating direction method of multipliers, Found. Trends® Mach.Learn. 3 (1) (2011) 1–122.

[37] H. Zhang, Z. Lin, C. Zhang, E.Y. Chang, Exact recoverability of robust PCA via outlier pursuit with tight recovery bounds, AAAI, 2015.

[38] Y. Peng, B.-L. Lu, Discriminative extreme learning machine with supervised sparsity preserving for image classification, Neurocomputing 261 (2017) 242–252.

[39] Y. Peng, B.-L. Lu, Robust structured sparse representation via half-quadratic optimization for face recognition, Multimedia Tool. Appl. 76 (6) (2017) 8859–8880.