Journal of Scientific Computing (2021) 89:67
https://doi.org/10.1007/s10915-021-01679-6

The Nonconvex Tensor Robust Principal Component Analysis Approximation Model via the Weighted ℓ_p-Norm Regularization

Minghui Li[1] · Wen Li[1] · Yannan Chen[1] · Mingqing Xiao[2]

Received: 3 March 2021 / Revised: 4 July 2021 / Accepted: 16 July 2021 /
Published online: 13 November 2021

Abstract

Tensor robust principal component analysis (TRPCA), which aims to recover the underlying low-rank multidimensional datasets from observations corrupted by noise and/or outliers, has been widely applied to various fields. The typical convex relaxation of TRPCA in literature is to minimize a weighted combination of the tensor nuclear norm (TNN) and the ℓ_1-norm. However, owing to the gap between the tensor rank function and its lower convex envelop (i.e., TNN), the tensor rank approximation by using the TNN appears to be insufficient. Also, the ℓ_1-norm generally is too relaxing as an estimator for the ℓ_0-norm to obtain desirable results in terms of sparsity. Different from current approaches in literature, in this paper, we develop a new non-convex TRPCA model, which minimizes a weighted combination of non-convex tensor rank approximation function and the weighted ℓ_p-norm to attain a tighter approximation. The resultant non-convex optimization model can be solved efficiently by the alternating direction method of multipliers (ADMM). We prove that the constructed iterative sequence generated by the proposed algorithm converges to a critical point of the proposed model. Numerical experiments for both image recovery and surveillance video background modeling demonstrate the effectiveness of the proposed method.

This research was supported by the National Natural Science Foundations of China (12071159, 11671158, U1811464) and the NSF-DMS 1854638 of the United States.

✉ Wen Li
liwen@scnu.edu.cn

Minghui Li
limh2016@126.com

Yannan Chen
ynchen@scnu.edu.cn

Mingqing Xiao
mxiao@siu.edu

[1] School of Mathematical Sciences, South China Normal University, Guangzhou 510631, People's Republic of China

[2] Department of Mathematics, Southern Illinois University-Carbondale, Carbondale, IL 62901, USA

Keywords Tensor robust principal component analysis · Low rank approximation · Tensor singular value decomposition · Non-convex optimization · Alternating direction method of multipliers · ℓ_p-Norm

Mathematics Subject Classification 15A69 · 65K05

1 Introduction

With the rapid development of science and technology, more and more complex datasets are obtainable and appear to be multi-dimensional in nature [22,39]. For example, a color image, consisting of three color channels (red, green, blue), is a three-dimensional data. A greyscale video consisted of two pixel variables and a time variable is modeled as a three-dimensional dataset, while a color video comprised of color images is characterized as a four-dimensional dataset. Mathematically, multi-dimensional data can be formulated as a tensor [22], which is a natural generalization of a matrix. Employing tensor models to handle various multi-dimensional datasets has become increasingly valuable since the tensor representation maintains the spatial structure information among individual entries and provides a more suitable characterization for datasets. Quite often, many real-world multi-dimensional datasets can be approximated by low dimensional subspaces due to the underlying correlation inside the datasets [34]. Low rank properties not only greatly simplify the data representation but also allow us to process multi-dimensional datasets more efficiently. Therefore, the study of low-rank optimization problems under the tensor framework becomes not only important but also critical in many applications.

As one of the important applications of the low-rank matrix optimization, robust principal component analysis (RPCA) [4] has been widely used in many areas including subspace clustering [24], machine learning [7], face recognition [13], etc. In the past, common approaches by using RPCA for multi-dimensional datasets are to directly restructure the datasets into matrices or vectors. However, such approaches result in the loss of structural information that may be critical in data characterization. Recently, tensor robust principal component analysis (TRPCA) has been developed [25,31] to handle the multi-dimensional datasets. Suppose that we are given an observed tensor data $\mathcal{X} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ and know that it can be split as $\mathcal{X} = \mathcal{L} + \mathcal{E}$, where $\mathcal{L}$ and $\mathcal{E}$ are low rank and sparse parts of $\mathcal{X}$, respectively. TRPCA is used to extract $\mathcal{L}$ and $\mathcal{E}$ from $\mathcal{X}$, its mathematical model can be formulated as

$$\min_{\mathcal{L},\mathcal{E}} \quad \operatorname{rank}(\mathcal{L}) + \lambda \|\mathcal{E}\|_0, \quad \text{s.t.} \quad \mathcal{X} = \mathcal{L} + \mathcal{E}, \tag{1.1}$$

where λ is a positive regularization parameter used to balance the low rank and sparse terms, $\|\mathcal{E}\|_0$ denotes the number of nonzero entries of $\mathcal{E}$. The key to solve the above optimization problem is to use a suitable tensor rank characterization. However, one of the major challenges for the tensor lies in that the characterizations of the tensor rank are not always tractable. Based on problems at hand, many tensor rank definitions have been proposed [8,15,22,36] and have led to different tensor rank minimization models while each has its limitation and applicable scope. For example, the CANDECOMP/PARAFAC (CP) rank [15] of a tensor $\mathcal{X}$, defined as the minimum number of rank-one tensors required to express $\mathcal{X}$, is generally NP-hard to compute [40,53]. Although many approaches have successfully recovered some special low CP rank tensors (see e.g., [19]), it is often a challenge to determine the CP rank and its best convex relaxation. The Tucker rank of the Mth-order tensor $\mathcal{X}$ is defined to be an M-dimension vector whose i-th component is the rank of the mode-i unfolding matrix of $\mathcal{X}$ [8,22]. Liu et al. [25] introduced a tractable convex method for the Tucker rank tensor completion problem by defining the sum of the nuclear norms (SNN) of all unfolding matrices of the tensor. Numerical experiments on TRPCA by using SNN was presented in [29] by Lu et al. However, the solution of the above model appears to be suboptimal since SNN is not the convex envelope of the sum of ranks of all those unfolding matrices. The tensor train (TT) rank [36] is composed by ranks of matrices formed by a well-balanced matricization scheme, i.e., matricize the tensor along one or a few modes. Bengua et al. [1] developed a TT rank tensor completion method for color images and videos. Yang et al. [45] applied the TT nuclear norm (TTNN) to the TRPCA problem with good numerical results. However, the performance of the TT-based methods is always unsatisfactory when the third-dimension of the data is large (such as for the hyperspectral images).

Recently, the tensor singular value decomposition (t-SVD) developed by Kilmer et al. [20] has attracted considerable interests. Based on the definition of tensor-tensor product (denoted as " $*$ ") , t-SVD decomposes a tensor $\mathcal{X}$ into one f-diagonal tensor $\mathcal{S}$ multiplied by two orthogonal tensors $\mathcal{U}$ and $\mathcal{V}$: $\mathcal{X} = \mathcal{U} * \mathcal{S} * \mathcal{V}^T$. By using the discrete Fourier transform (DFT), the t-SVD can be easily obtained by computing certain matrix SVDs in the Fourier domain. With the framework of the t-SVD, Kilmer et al. provided a new tensor rank definition named tubal rank [21]. Compared with other tensor decompositions, the t-SVD has been shown to be more robust in maintaining the intrinsic correlative structure of the multi-dimensional data [29]. Based on the t-SVD, Lu et al. [29] proposed the following TRPCA model by replacing the tensor tubal rank with the tensor nuclear norm (TNN) (see Sect. 2 for the definition) and the ℓ_0-norm with the ℓ_1-norm:

$$\min_{\mathcal{L},\mathcal{E}} \|\mathcal{L}\|_* + \lambda\|\mathcal{E}\|_{\ell_1}, \quad \text{s.t.} \quad \mathcal{X} = \mathcal{L} + \mathcal{E}, \tag{1.2}$$

where $\|\mathcal{L}\|_*$ denotes the TNN of $\mathcal{L}$. However, this model has some important issues needed to address in applications. First, the gap between the tensor rank function and its lower convex envelop (i.e., TNN) may be large, especially when some of the singular values of the original tensor are very large. The TNN minimization method (1.2) usually penalizes the large singular values and leads to the loss of main information [14]. Also, the ℓ_1-norm is a coarse approximation of the ℓ_0-norm, which usually leads to undesirable results.

Inspired by the success of the non-convex matrix rank minimization models [18,37], many non-convex tensor rank minimization models have been proposed to improve the aforementioned approaches. For instance, Jiang et al. [16] proposed to use the partial sum of the tensor nuclear norm (PSTNN) as the approximation for the tensor tubal rank. Cai et al. [2] established a non-convex TRPCA model by introducing a t-Gamma tensor quasi-norm as a non-convex regularization to approximate the tensor tubal rank. Beyond TRPCA, many non-convex rank surrogates were proposed in the tensor completion (TC) problem, e.g., Laplace [43], Minimax Concave Penalty (MCP) [12,49], Smoothly Clipped Absolute Deviation (SCAD) [12,49], Exponential-Type Penalty (ETP) [12], and Logarithmic [42]. The advantage of non-convex approximation model is due to that it can capture the tensor rank more accurately than the convex setting and thus can obtain a more accurate result than convex approaches. For example, in [42], the logarithmic function based non-convex surrogate is shown to be a more accurate approximation for the Tucker rank. In [43], the laplace function based non-convex surrogate is shown to be a better measure of the tubal rank compared with TNN.

In this paper, we mainly focus on the non-convex approach for the TRPCA problem. Different from existing non-convex TRPCA models, we not only propose a non-convex approximation for the tensor tubal rank but also develop a new and suitable non-convex characterization for the sparse constraint term. The properties and advantages of the proposed two non-convex relaxations will be illustrated in Sect. 3 with more details. The main contributions of this paper are summarized as follows:

(a) In this paper, we develop a feasible non-convex TRPCA model, in which a non-convex e^{γ}-type approximation rank function is established for a more accurate rank approximation under the tensor framework, while at the same time, a weighted ℓ_p-norm is developed being better capture the sparsity of the noise term without requiring p to be too small.
(b) The corresponding alternating direction method of multipliers (ADMM) is constructed to solve the optimization model, which can be divided into two sub-problems, in which two sub-problems can be solved effectively by the generalized soft-thresholding (GST) operator and local linearization minimization approach, respectively. We show that the sequence generated by the proposed algorithm converges to a KKT stationary point.
(c) Extensive numerical experiments with several benchmark datasets are given to demonstrate the robustness and the efficiency of the proposed model, compared with some related existing approaches.

The rest of the paper is organized as follows. Related notations and definitions are given in Sect. 2. In Sect. 3, the proposed non-convex TRPCA model is presented. Meanwhile, based on the ADMM algorithm, we provide a detailed process for solving the proposed model. In Sect. 4, we analyze the convergence of the proposed algorithm. The numerical results are shown in Sect. 5. The paper ends with concluding remarks in Sect. 6.

2 Preliminaries

In this section, we first introduce some basic tensor notations used in this paper and then present the t-SVD algebraic framework. More details about tensors and the t-SVD can be found in [20,38].

2.1 Notation

The fields of real numbers and complex numbers are denoted by $\mathbb{R}$ and $\mathbb{C}$, respectively. Throughout this paper, scalars, vectors, matrices, and tensors are denoted by lowercase letters (e.g., x), boldface lowercase letters (e.g., $\mathbf{x}$), boldface capital letters (e.g., $\mathbf{X}$), and boldface calligraphy letters (e.g., $\mathcal{X}$), respectively.

For a third-order tensor $\mathcal{X}$, its (i, j, k)-th entry is denoted by $\mathcal{X}(i, j, k)$ or $\mathcal{X}_{ijk}$. We use Matlab notations $\mathcal{X}(i, :, :)$, $\mathcal{X}(:, i, :)$, and $\mathcal{X}(:, :, i)$ to denote the i-th horizontal, lateral, and frontal slices of $\mathcal{X}$, respectively. For simplicity, $\mathcal{X}(:, :, i)$ is denoted by $\mathbf{X}_i$. The (i, j)-th tube of $\mathcal{X}$, denoted by the Matlab notation $\mathcal{X}(i, j, :)$, is a vector obtained by fixing the first two indices and varying the third one. The inner product of two same size tensors $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{N_1\times N_2\times N_3}$ is defined as $\langle \mathcal{X}, \mathcal{Y}\rangle := \sum_{i=1}^{N_1}\sum_{j=1}^{N_2}\sum_{k=1}^{N_3} \mathcal{X}_{ijk}\mathcal{Y}_{ijk}$. The Frobenius norm, the ℓ_1-norm, and the infinity norm of a third-order tensor $\mathcal{X}$ are defined as $\|\mathcal{X}\|_F := (\sum_{i,j,k} |\mathcal{X}_{ijk}|^2)^{\frac{1}{2}}$, $\|\mathcal{X}\|_{\ell_1} := \sum_{i,j,k} |\mathcal{X}_{ijk}|$, and $\|\mathcal{X}\|_\infty := \max_{i,j,k} |\mathcal{X}_{ijk}|$, respectively.

For a third-order tensor $\mathcal{X}$, we use $\hat{\mathcal{X}} = \text{fft}(\mathcal{X}, [\,], 3)$ to denote the DFT along the third dimension (by using the Matlab command fft).

It is noted that the $N_3 \times N_3$ DFT matrix $\mathbf{F}_{N_3}$ is not unitary, but $\tilde{\mathbf{F}}_{N_3} = \frac{1}{\sqrt{N_3}}\mathbf{F}_{N_3}$ is unitary. In later theoretical analysis we see that $\mathbf{F}_{N_3}$ may be replaced by the unitary matrix $\tilde{\mathbf{F}}_{N_3}$.

2.2 Review of the t-SVD

The t-SVD was first proposed by Kilmer et al. [20] and has been widely used in TC [17,28, 35,41,43,47–49,51,52], image multi-view subspace clustering [46] and TRPCA [2,6,29].

For a third-order tensor $\mathcal{X} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, five block-based operations bcirc$(\cdot)$, unfold$(\cdot)$, fold$(\cdot)$, bdiag$(\cdot)$ and unbdiag$(\cdot)$ are defined as follows:

$$\mathrm{bcirc}(\mathcal{X}) = \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_{N_3} & \dots & \mathbf{X}_2 \\ \mathbf{X}_2 & \mathbf{X}_1 & \dots & \mathbf{X}_3 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{X}_{N_3} & \mathbf{X}_{N_3-1} & \dots & \mathbf{X}_1 \end{bmatrix},$$

$$\mathrm{unfold}(\mathcal{X}) = \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \vdots \\ \mathbf{X}_{N_3} \end{bmatrix}, \quad \mathrm{fold}(\mathrm{unfold}(\mathcal{X})) = \mathcal{X},$$

$$\mathrm{bdiag}(\mathcal{X}) = \begin{bmatrix} \mathbf{X}_1 & & & \\ & \mathbf{X}_2 & & \\ & & \ddots & \\ & & & \mathbf{X}_{N_3} \end{bmatrix}, \quad \mathrm{unbdiag}(\mathrm{bdiag}(\mathcal{X})) = \mathcal{X}.$$

Definition 1 (*t-product* [20]) The t-product $\mathcal{X} * \mathcal{Y}$ of $\mathcal{X} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ and $\mathcal{Y} \in \mathbb{R}^{N_2 \times N_4 \times N_3}$ is a tensor $\mathcal{Z} \in \mathbb{R}^{N_1 \times N_4 \times N_3}$ given by

$$\mathcal{Z} = \mathcal{X} * \mathcal{Y} = \mathrm{fold}(\mathrm{bcirc}(\mathcal{X})\, \mathrm{unfold}(\mathcal{Y})). \tag{2.1}$$

Remark 1 Different from direct unfolding the tensor $\mathcal{X}$ along the third mode, the block circulant matrix bcirc$(\mathcal{X})$ of $\mathcal{X}$ keeps the order of frontal slices of $\mathcal{X}$ in a circulant way that maintains the structure of $\mathcal{X}$ in terms of the frontal direction.

It is well known that the circulant matrix can be diagonalized by DFT [20], Kilmer et al. gave a similar property for block-circulant matrix as follows.

Property 1 [20] *For a tensor $\mathcal{X} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, its block circulant matrix* bcirc$(\mathcal{X})$ *can be block-diagonalized by DFT as follows:*

$$\mathrm{bdiag}(\hat{\mathcal{X}}) = \left(\tilde{\mathbf{F}}_{N_3} \otimes \mathbf{I}_{N_1}\right) \mathrm{bcirc}(\mathcal{X}) \left(\tilde{\mathbf{F}}^*_{N_3} \otimes \mathbf{I}_{N_2}\right), \tag{2.2}$$

where $\tilde{\mathbf{F}}_{N_3} = \frac{1}{\sqrt{N_3}}\mathbf{F}_{N_3}$, $\mathbf{F}_{N_3}$ is an $N_3 \times N_3$ DFT matrix, $\mathbf{F}^*_{N_3}$ denotes its conjugate transpose, the notation $\otimes$ represents the Kronecker product, and $\mathbf{I}_N$ is an $N \times N$ identity matrix.

Remark 2 From Property 1, t-product $\mathcal{Z} = \mathcal{X} * \mathcal{Y}$ in (2.1) implies

$$\begin{aligned} \mathrm{unfold}(\mathcal{Z}) &= \mathrm{bcirc}(\mathcal{X})\, \mathrm{unfold}(\mathcal{Y}) \\ &= (\tilde{\mathbf{F}}^*_{N_3} \otimes \mathbf{I}_{N_1}) \left((\tilde{\mathbf{F}}_{N_3} \otimes \mathbf{I}_{N_1})\, \mathrm{bcirc}(\mathcal{X})(\tilde{\mathbf{F}}^*_{N_3} \otimes \mathbf{I}_{N_2})\right) \left((\tilde{\mathbf{F}}_{N_3} \otimes \mathbf{I}_{N_2})\, \mathrm{unfold}(\mathcal{Y})\right) \\ &= (\tilde{\mathbf{F}}^*_{N_3} \otimes \mathbf{I}_{N_1})\, \mathrm{bdiag}(\hat{\mathcal{X}})\, \mathrm{unfold}(\hat{\mathcal{Y}}). \end{aligned} \tag{2.3}$$

Then left multiplying both sides of (2.3) by $\tilde{\mathbf{F}}_{N_3} \otimes \mathbf{I}_{N_1}$ gives

$$\text{unfold}(\hat{\mathcal{Z}}) = \text{bdiag}(\hat{\mathcal{X}})\,\text{unfold}(\hat{\mathcal{Y}}). \tag{2.4}$$

The equation (2.4) is equivalent to

$$\hat{\mathbf{Z}}_k = \hat{\mathbf{X}}_k \hat{\mathbf{Y}}_k, \quad k = 1, \ldots, N_3, \tag{2.5}$$

which means the t-product (2.1) in the original domain can be computed by the matrix multiplication of the frontal slices in the Fourier domain. This greatly simplifies the computational procedure.

Remark 3 [29] Since $\tilde{\mathbf{F}}_{N_3} \otimes \mathbf{I}_{N_1}$ and $\tilde{\mathbf{F}}^*_{N_3} \otimes \mathbf{I}_{N_2}$ in (2.2) are unitary matrices, it follows from (2.2) that

$$\|\mathcal{X}\|_F^2 = \frac{1}{N_3} \|\,\text{bdiag}(\hat{\mathcal{X}})\|_F^2.$$

Definition 2 (*Identity tensor and f-diagonal tensor* [20]) The identity tensor $\mathcal{I} \in \mathbb{R}^{N_1 \times N_1 \times N_3}$ is the tensor whose first frontal slice is an identity matrix and other frontal slices are all zeros. A tensor is called f-diagonal if each frontal slice is a diagonal matrix.

Definition 3 (*Tensor transpose and orthogonal tensor* [20]) The transpose of a tensor $\mathcal{X} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ is the tensor $\mathcal{X}^T \in \mathbb{R}^{N_2 \times N_1 \times N_3}$ obtained by transposing each frontal slice of $\mathcal{X}$ and then reversing the order of transposed frontal slices 2 through N_3, i.e., $\text{unfold}(\mathcal{X}^T) = [\mathbf{X}_1, \mathbf{X}_{N_3}, \cdots, \mathbf{X}_2]^T$. A tensor $\mathcal{Q} \in \mathbb{R}^{N_1 \times N_1 \times N_3}$ is orthogonal if it satisfies $\mathcal{Q} * \mathcal{Q}^T = \mathcal{Q}^T * \mathcal{Q} = \mathcal{I}$.

Based on the above definitions, we now restate the factorization strategy [20] for the third-order tensor.

Lemma 1 (t-SVD) *For $\mathcal{X} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, the t-SVD of $\mathcal{X}$ is given by*

$$\mathcal{X} = \mathcal{U} * \mathcal{S} * \mathcal{V}^T, \tag{2.6}$$

where $\mathcal{U} \in \mathbb{R}^{N_1 \times N_1 \times N_3}$, $\mathcal{V} \in \mathbb{R}^{N_2 \times N_2 \times N_3}$ are orthogonal tensors, and $S \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ is an f-diagonal tensor.

Remark 4 According to Remark 2, we have $\hat{\mathbf{X}}_k = \hat{\mathbf{U}}_k \hat{\mathbf{S}}_k \hat{\mathbf{V}}_k^T$ for $k = 1, \ldots, N_3$, where $\hat{\mathbf{X}}_k$, $\hat{\mathbf{U}}_k$, $\hat{\mathbf{S}}_k$ and $\hat{\mathbf{V}}_k$ are frontal slices of $\hat{\mathcal{X}}, \hat{\mathcal{U}}, \hat{\mathcal{S}}$ and $\hat{\mathcal{V}}$, respectively. This means the t-SVD can be obtained by computing several matrix SVDs in the Fourier domain.

Definition 4 (*Tensor tubal rank* [21]) Given a tensor $\mathcal{X} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, its tubal rank is defined as the number of nonzero singular tubes of the f-diagonal tensor $\mathcal{S}$, i.e.,

$$\text{rank}_t(\mathcal{X}) = \#\left\{i, \mathcal{S}(i,i,:) \neq 0\right\}, \tag{2.7}$$

where $\mathcal{S}$ comes from the t-SVD of $\mathcal{X} = \mathcal{U} * \mathcal{S} * \mathcal{V}^T$ and the notation "#" denotes the cardinality of a set.

Definition 5 (*T-SVD based TNN* [29]) The TNN of $\mathcal{X} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ is given by

$$\|\mathcal{X}\|_* = \frac{1}{N_3} \sum_{k=1}^{N_3} \|\hat{\mathbf{X}}_k\|_* = \frac{1}{N_3} \sum_{k=1}^{N_3} \sum_{i=1}^{\min\{N_1, N_2\}} \sigma_i(\hat{\mathbf{X}}_k), \tag{2.8}$$

where $\sigma_i(\hat{\mathbf{X}}_k)$ is the i-th singular value of matrix $\hat{\mathbf{X}}_k$ and $\sigma_1(\hat{\mathbf{X}}_k) \geq \cdots \geq \sigma_{\min\{N_1,N_2\}}(\hat{\mathbf{X}}_k)$ for all $k = 1, \ldots, N_3$.

3 The TRPCA Model and Algorithms

In this section, we first give the motivation and then propose a non-convex TRPCA model based on some non-convex functions. Finally, an algorithm for solving the model is established.

3.1 Motivation

- **Motivation for the low rank part:**

In this paper, to overcome the disadvantages of the model (1.2), we propose to use a non-convex but smooth e^γ-type function to approximate the tensor tubal rank. The e^γ-type function used in [37] is for the matrix rank approximation, and we will adopt it under t-SVD framework for tensor rank approximation. The e^γ-type function is given by

$$\phi(x) = \frac{e^\gamma x}{\gamma + x}, \quad x \in [0, +\infty), \tag{3.1}$$

where γ is a positive parameter. It can be seen that ϕ is a smooth, monotonously increasing and concave function. Using the e^γ-type function $\phi(x)$, a non-convex surrogate for an $M \times N$ matrix $\mathbf{X}$ is given by [37]:

$$\|\mathbf{X}\|_{e^\gamma} := \sum_{i=1}^{\min\{M,N\}} \phi(\sigma_i(\mathbf{X})), \quad \gamma > 0, \tag{3.2}$$

where $\sigma_i(\mathbf{X})$ is the i-th singular value of $\mathbf{X}$ and $\sigma_1(\mathbf{X}) \geq \cdots \geq \sigma_{\min\{M,N\}}(\mathbf{X})$. For a third-order tensor $\mathcal{X} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, we define a non-convex surrogate for the tubal rank as

$$\|\mathcal{X}\|_{t-e^\gamma} := \frac{1}{N_3} \sum_{k=1}^{N_3} \|\hat{\mathbf{X}}_k\|_{e^\gamma}, \quad \gamma > 0, \tag{3.3}$$

where $\hat{\mathbf{X}}_k$ is the k-th frontal slice of $\hat{\mathcal{X}} = \text{fft}(\mathcal{X}, [\,], 3) = [\hat{\mathbf{X}}_1 | \cdots | \hat{\mathbf{X}}_{N_3}]$ obtained by applying FFT on $\mathcal{X}$ along the third mode.

Now we explain why we adopt the e^γ-type function as the non-convex surrogate for the tubal rank. We randomly select twenty color images from Berkeley Segmentation Dataset (BSD) [33], the size of each image is $321 \times 481 \times 3$ or $481 \times 321 \times 3$. For each image, in Fig. 1a, we show the comparison of associated results coming from the tensor tubal rank given by (2.7), the TNN given by (2.8), and the e^γ-type approximation rank ($\gamma = 1.1$) given by (3.3). We also illustrate the distance between $\|\cdot\|_{t-e^\gamma}$, $\|\cdot\|_*$ and the tubal rank for each image in Fig. 1b. It can be seen from Fig. 1 that the result obtained by (3.3) gives a tighter approximation to the tensor tubal rank than that obtained by the TNN for each image. The above comparison indicates that our proposed non-convex surrogate (3.3) is a more accurate approximation of the tensor tubal rank.

The e^γ-type non-convex rank approximation function (3.3) has the following property.

Property 2 *Let $\mathcal{X} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$. The e^γ-type rank surrogate function (3.3) is unitarily invariant, i.e., $\|\mathcal{U} * \mathcal{X} * \mathcal{V}^T\|_{t-e^\gamma} = \|\mathcal{X}\|_{t-e^\gamma}$, where $\mathcal{U} \in \mathbb{R}^{N_1 \times N_1 \times N_3}$ and $\mathcal{V} \in \mathbb{R}^{N_2 \times N_2 \times N_3}$ are orthogonal tensors. In addition, $\lim_{\gamma \to 0} \|\mathcal{X}\|_{t-e^\gamma} = \text{rank}_t(\mathcal{X})$.*

Proof The proof of $\lim_{\gamma \to 0} \|\mathcal{X}\|_{t-e^\gamma} = \text{rank}_t(\mathcal{X})$ is straightforward, we omit it here. Now we show that the e^γ-type rank surrogate function (3.3) is unitarily invariant.

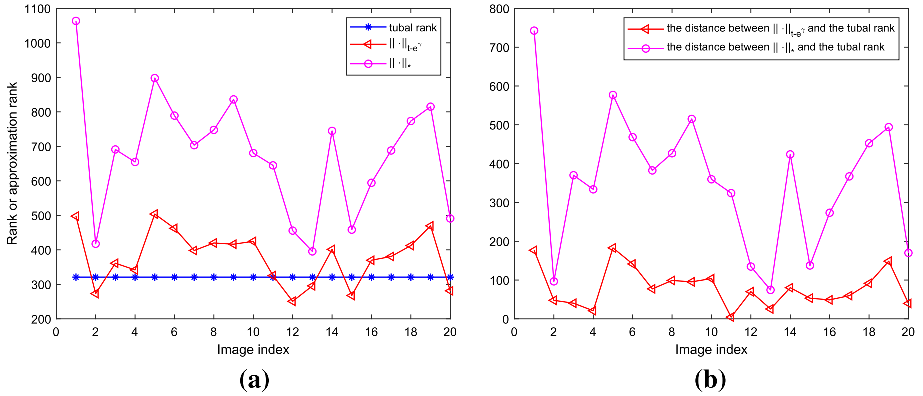


Fig. 1 Comparison (3.3) with (2.7) and (2.8)

Let $\mathcal{Z} \in \mathbb{R}^{n\times n\times m}$ be an orthogonal tensor. We first show that all frontal slices $\hat{\mathbf{Z}}_k$ $(k = 1,\ldots,m)$ of $\hat{\mathcal{Z}}$ are orthogonal matrices. Since $\mathcal{Z} * \mathcal{Z}^T = \mathcal{Z}^T * \mathcal{Z} = \mathcal{I}$, it follows from Remark 2 that $\hat{\mathbf{Z}}_k\hat{\mathbf{Z}}_k^T = \hat{\mathbf{Z}}_k^T\hat{\mathbf{Z}}_k = \hat{\mathbf{I}}_k$ for $k = 1,\ldots,m$. Because $\mathcal{I}$ is an identity tensor, according to the property of the Fourier transform, it is easy to see that $\hat{\mathbf{I}}_k$ is an identity matrix for $k = 1,\ldots,m$. Hence, $\hat{\mathbf{Z}}_k$ is an orthogonal matrix for $k = 1,\ldots,m$.

According to Remark 2 and (3.3), we have

$$\|\mathcal{U} * \mathcal{X} * \mathcal{V}^T\|_{t-e^\gamma} = \frac{1}{N_3}\sum_{k=1}^{N_3}\|\hat{\mathbf{U}}_k\hat{\mathbf{X}}_k\hat{\mathbf{V}}_k^T\|_{e^\gamma}. \tag{3.4}$$

Since $\mathcal{U}$ and $\mathcal{V}$ are orthogonal tensors, it follows that $\hat{\mathbf{U}}_k$ and $\hat{\mathbf{V}}_k$ are orthogonal matrices for $k = 1,\ldots,N_3$. Therefore, by (3.2) and (3.3) we have

$$\frac{1}{N_3}\sum_{k=1}^{N_3}\|\hat{\mathbf{U}}_k\hat{\mathbf{X}}_k\hat{\mathbf{V}}_k^T\|_{e^\gamma} = \frac{1}{N_3}\sum_{k=1}^{N_3}\|\hat{\mathbf{X}}_k\|_{e^\gamma} = \|\mathcal{X}\|_{t-e^\gamma}.$$

This proves the conclusion. □

- **Motivation for the sparse constraint term:**

It is well known that the ℓ_0-norm in the sparse constraint term is usually relaxed by using the ℓ_1-norm due to its simplicity. The obtained solution by the ℓ_1-norm minimization is often suboptimal to the original ℓ_0-norm minimization since the ℓ_1-norm is a coarse approximation of the ℓ_0-norm. Some non-convex surrogates of ℓ_0-norm have been applied in the fields of signal and image processing to find the sparsest solution (see e.g., [5,44]), but the non-convex surrogates of ℓ_0-norm in the sparse constraint term in tensor case have not been well studied. Recently, ℓ_p-norm $(0 < p < 1)$ was proposed to as an approximation for the ℓ_0-norm by using small positive p in TRPCA problem [6]. Numerical tests given by [6] showed that non-convex ℓ_p-norm outperformed ℓ_1-norm. However, small p will lead to the insensitivity to tensor elements, for example, $1000^p \approx 0.1^p$ when p is small enough, and larger elements will be misclassified as small entries in later optimization process (e.g., see Step 1 of (3.8)). Inspired by the idea that sparsity enhancement by using the weighted ℓ_1-norm minimization in signal recovery [3,50], we propose to use the weighted ℓ_p-norm as the non-convex relaxation of the ℓ_0-norm. Given a tensor $\mathcal{X} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, the weighted ℓ_p-norm for $\mathcal{X}$ is given by

$$\|\mathcal{X}\|_{\mathcal{W},\ell_p} := \left(\sum_{i,j,k} \left(\mathcal{W}_{ijk} |\mathcal{X}_{ijk}| \right)^p \right)^{\frac{1}{p}}, \quad 0 < p < 1, \tag{3.5}$$

where $\mathcal{W}$ is a nonnegative weight tensor. It is not difficult to see

$$\lim_{p \to 1} \lim_{\epsilon \to 0} \lim_{\mathcal{W}_{ijk} \to \frac{1}{|\mathcal{X}_{ijk}|+\epsilon}} \|\mathcal{X}\|_{\mathcal{W},\ell_p} = \|\mathcal{X}\|_0.$$

This clearly implies that the introduction of the weighted ℓ_p-norm (3.5) can allow p not necessarily to be too small but maintain the sparse characteristics.

3.2 The Proposed Model

Equipped with the two non-convex surrogates given by (3.3) and (3.5), we establish the following non-convex TRPCA model:

$$\min_{\mathcal{L},\mathcal{E}} \|\mathcal{L}\|_{t-e^\gamma} + \lambda \|\mathcal{E}\|_{\mathcal{W},\ell_p}^p, \quad \text{s.t.} \quad \mathcal{X} = \mathcal{L} + \mathcal{E}, \tag{3.6}$$

where the weight tensor $\mathcal{W}$ will be updated in each iteration and the (i, j, k)-th entry of $\mathcal{W}$ in the $(k+1)$-th iteration is $\dfrac{1}{|\mathcal{E}_{ijk}^{(k)}| + \epsilon}$, here $\mathcal{E}_{ijk}^{(k)}$ is the (i, j, k)-th entry of $\mathcal{E}$ in the k-th iteration and ϵ is a small positive constant.

We develop the ADMM to solve the proposed model (3.6). The augmented Lagrange function of (3.6) is given by

$$\begin{aligned} L_\beta(\mathcal{L}, \mathcal{E}, \mathcal{M}) &= \|\mathcal{L}\|_{t-e^\gamma} + \lambda \, \|\mathcal{E}\|_{\mathcal{W},\ell_p}^p - \langle \mathcal{X} - \mathcal{L} - \mathcal{E}, \mathcal{M} \rangle + \frac{\beta}{2} \|\mathcal{X} - \mathcal{L} - \mathcal{E}\|_F^2 \\ &= \|\mathcal{L}\|_{t-e^\gamma} + \lambda \, \|\mathcal{E}\|_{\mathcal{W},\ell_p}^p + \frac{\beta}{2} \|\mathcal{X} - \mathcal{L} - \mathcal{E} - \frac{\mathcal{M}}{\beta}\|_F^2 - \frac{1}{2\beta} \langle \mathcal{M}, \mathcal{M} \rangle, \end{aligned} \tag{3.7}$$

where $\mathcal{M}$ is the Lagrange multiplier and β is the penalty parameter. According to the framework of ADMM, $\mathcal{L}$, $\mathcal{E}$, and $\mathcal{M}$ are alternately updated as:

$$\begin{cases} Step\ 1: \mathcal{E}_{k+1} = \arg\min_{\mathcal{E}} \lambda \|\mathcal{E}\|_{\mathcal{W}_k,\ell_p}^p + \frac{\beta_k}{2} \|\mathcal{B}_k - \mathcal{E}\|_F^2 \quad \text{with} \quad \mathcal{B}_k = \mathcal{X} - \mathcal{L}_k - \frac{\mathcal{M}_k}{\beta_k}, \\ Step\ 2: \mathcal{L}_{k+1} = \arg\min_{\mathcal{L}} \frac{1}{\beta_k} \|\mathcal{L}\|_{t-e^\gamma} + \frac{1}{2} \|\mathcal{A}_k - \mathcal{L}\|_F^2 \quad \text{with} \quad \mathcal{A}_k = \mathcal{X} - \mathcal{E}_{k+1} - \frac{\mathcal{M}_k}{\beta_k}, \\ Step\ 3: \mathcal{M}_{k+1} = \mathcal{M}_k + \beta_k (\mathcal{L}_{k+1} + \mathcal{E}_{k+1} - \mathcal{X}). \end{cases} \tag{3.8}$$

1) Update $\mathcal{E}$: Before giving the solution of **Step 1**, we take a look at the ℓ_p-norm minimization problem for scalars which was given in [54].

Lemma 2 [54] *For the given p $(0 < p < 1)$ and $\lambda > 0$, an optimal solution of the following optimization problem*

$$\arg\min_x \frac{1}{2}(y - x)^2 + \lambda |x|^p \tag{3.9}$$

is given by the generalized soft-thresholding (GST) operator, which is defined as

$$GST(y, \lambda, p) = \begin{cases} 0, & \textit{if } |y| \le \tau_p^{GST}(\lambda), \\ \operatorname{sgn}(y) \cdot S_p^{GST}(y, \lambda), & \textit{if } |y| > \tau_p^{GST}(\lambda), \end{cases}$$

where $\operatorname{sgn}(y)$ *denotes the signum function,* $\tau_p^{GST}(\lambda) = (2\lambda(1-p))^{\frac{1}{2-p}} + \lambda p\,(2\lambda(1-p))^{\frac{p-1}{2-p}}$ *is a threshold value, and* $S_p^{GST}(y,\lambda)$ *can be obtained by solving the following equation*

$$S_p^{GST}(y,\lambda) - y + \lambda p (S_p^{GST}(y,\lambda))^{p-1} = 0. \tag{3.10}$$

The iterative algorithm for solving (3.9) is summarized in Algorithm 1 [54].

Algorithm 1 Generalized Soft-Thresholding (GST) [54]

```
Input: $y, \lambda, p, J = 2$ or 3.
$\tau_p = (2\lambda(1-p))^{\frac{1}{2-p}} + \lambda p\,(2\lambda(1-p))^{\frac{p-1}{2-p}}$
if $|y| \leq \tau_p$ then
    $\mathrm{GST}(y, \lambda, p) = 0$
else
    $k = 0, x^k = |y|$
    for $k = 0, 1, \ldots, J$ do
        $x^{(k+1)} = |y| - \lambda p (x^{(k)})^{p-1}$
        $k = k + 1$
    end for
    $\mathrm{GST}(y, \lambda, p) = \operatorname{sgn}(y) x^{(k)}$
end if
Output: $\mathrm{GST}(y, \lambda, p)$
```

According to **Step 1**, the $(k+1)$-th iteration is

$$\mathcal{E}_{ijk}^{(k+1)} = \underset{\mathcal{E}_{ijk}}{\arg\min} \lambda \left(\mathcal{W}_{ijk}^{(k)} |\mathcal{E}_{ijk}|\right)^p + \frac{\beta_k}{2}\left(b_{ijk}^{(k)} - \mathcal{E}_{ijk}\right)^2.$$

From Lemma 2, the solution with respect to $\mathcal{E}_{k+1}$ is given by

$$\mathrm{GST}\left(\mathcal{B}_k, \frac{\lambda \mathcal{W}_k^p}{\beta_k}, p\right), \tag{3.11}$$

where $\mathcal{W}_k^p$ is a weight tensor whose (i, j, k)-th entry is $\left(\frac{1}{|\mathcal{E}_{ijk}^{(k)}| + \epsilon}\right)^p$, ϵ is a small positive constant.

2) Update $\mathcal{L}$: We first give the global convergence results for the descent algorithm [30], which will be used to analyze the convergence of the fixed point inner iteration in Theorem 1.

Lemma 3 (Global Convergent Theorem in Chapter 7.7 of [30]) *Let A be an algorithm on a real finite-dimensional metric space X and suppose that, for a given x_0, the sequence $\{x_k\}_{k=0}^{\infty}$ is generated by $x_{k+1} \in A(x_k)$. Let a solution set $\Gamma \subset X$ be given and suppose*

(i) all points x_k are contained in a compact set $S \subset X$,
(ii) there is a continuous function Z on X such that if $x \notin \Gamma$ then $Z(y) < Z(x)$ for all $y \in A(x)$; if $x \in \Gamma$ then $Z(y) \leq Z(x)$ for all $y \in A(x)$,
(iii) the mapping A is closed at points outside Γ,

then the limit of any convergent subsequence of x_k is a solution.

The solution of the $\mathcal{L}$-subproblem is given by the following theorem.

Theorem 1 *Let $\mathcal{A}_k = \mathcal{U}_k * \Sigma_k * \mathcal{V}_k^T$ be the t-SVD of $\mathcal{A}_k \in \mathbb{R}^{N_1 \times N_2 \times N_3}$. Then an optimal solution of the following minimization*

$$\arg\min_{\mathcal{L}} \frac{1}{2}\|\mathcal{A}_k - \mathcal{L}\|_F^2 + \tau\|\mathcal{L}\|_{t-e^\gamma}, \tag{3.12}$$

is given by $\mathcal{L}^* = \mathcal{U}_k * \mathcal{S}_k * \mathcal{V}_k^T$, where $\mathcal{S}_k \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ is an f-diagonal tensor and $\tau = \frac{1}{\beta_k}$. And let $\hat{\Sigma}_k = [\hat{\Sigma}_{n,n,i}^{(k)}] = \text{fft}(\Sigma_k, [\ ], 3)$, $\hat{\mathcal{S}}_k = [\hat{S}_{n,n,i}^{(k)}] = \text{fft}(\mathcal{S}_k, [\ ], 3)$, $1 \le n \le \min\{N_1, N_2\}$, $1 \le i \le N_3$. Then $\hat{S}_{n,n,i}^{(k)}$ is the limit point of the following fixed point inner iteration

$$\hat{S}_{n,n,i}^{(k,l+1)} = \left(\hat{\Sigma}_{n,n,i}^{(k)} - \tau\frac{\gamma e^\gamma}{(\gamma + \hat{S}_{n,n,i}^{(k,l)})^2}\right)_+, \tag{3.13}$$

where $(x)_+ = \max\{x, 0\}$ and $\hat{S}_{n,n,i}^{(0,0)} = 0$.

Proof According to the Remark 3 and (3.3), the optimization problem (3.12) is equivalent to

$$\arg\min_{\hat{\mathbf{L}}_{k,i}} \frac{1}{N_3}\left(\sum_{i=1}^{N_3}\left(\frac{1}{2}\|\hat{\mathbf{A}}_{k,i} - \hat{\mathbf{L}}_{k,i}\|_F^2 + \tau\|\hat{\mathbf{L}}_{k,i}\|_{e^\gamma}\right)\right)$$

$$\Leftrightarrow \arg\min_{\hat{\mathbf{L}}_{k,i}} \frac{1}{2}\|\hat{\mathbf{A}}_{k,i} - \hat{\mathbf{L}}_{k,i}\|_F^2 + \tau\|\hat{\mathbf{L}}_{k,i}\|_{e^\gamma}, \quad i = 1, \ldots, N_3, \tag{3.14}$$

where $\hat{\mathbf{A}}_{k,i}$ and $\hat{\mathbf{L}}_{k,i}$ denote the i-th frontal slice of $\hat{\mathcal{A}}_k$ and $\hat{\mathcal{L}}_k$, respectively. That is to say, the tensor optimization problem (3.12) in original domain is transformed to N_3 matrix optimization problems (3.14) in Fourier domain.

Since $\mathcal{A}_k = \mathcal{U}_k * \Sigma_k * \mathcal{V}_k^T$, by Remark 4, we have $\hat{\mathbf{A}}_{k,i} = \hat{\mathbf{U}}_{k,i}\hat{\mathbf{\Sigma}}_{k,i}\hat{\mathbf{V}}_{k,i}^T$ for $i = 1, \ldots, N_3$. If $\hat{\mathbf{L}}_i^*$ minimize (3.14), by the von Neumann trace inequality for singular values (see Theorem 1 of [18]), we have

$$\hat{\mathbf{L}}_i^* = \hat{\mathbf{U}}_{k,i}\hat{\mathbf{S}}_{k,i}\hat{\mathbf{V}}_{k,i}^T, \quad i = 1, \ldots, N_3, \tag{3.15}$$

where $\hat{\mathbf{S}}_{k,i} = [\hat{S}_{n,n,i}^{(k)}]$ is the i-th frontal slice of $\hat{\mathcal{S}}_k$, and

$$\hat{S}_{n,n,i}^{(k)} = \arg\min_{\sigma \ge 0} \frac{1}{2}(\sigma - \hat{\Sigma}_{n,n,i}^{(k)})^2 + \tau\frac{e^\gamma \sigma}{\gamma + \sigma}. \tag{3.16}$$

Since the objective function in (3.16) is a combination of concave and convex functions, following the DC programming algorithm [10,18], we use a local minimization method to solve it. That is to say, we iteratively optimize (3.16) by linearizing the concave term $\phi(\sigma) = \dfrac{e^\gamma \sigma}{\gamma + \sigma}$ at each iteration, i.e., for the l-th inner iteration, we use the following first-order Taylor expansion (with Big "O" truncation error) to approximate $\phi(\sigma)$:

$$\phi(\sigma) = \phi(\hat{S}_{n,n,i}^{(k,l)}) + \frac{\gamma e^\gamma}{(\gamma + \hat{S}_{n,n,i}^{(k,l)})^2}(\sigma - \hat{S}_{n,n,i}^{(k,l)}) + O((\sigma - \hat{S}_{n,n,i}^{(k,l)})^2),$$

where $\hat{S}_{n,n,i}^{(k,l)}$ is the solution of (3.16) obtained in the l-th inner iteration and $\hat{S}_{n,n,i}^{(0,0)} = 0$. Therefore, (3.16) can be solved by iteratively solving the following optimization problem:

$$\hat{S}_{n,n,i}^{(k,l+1)} = \arg\min_{\sigma > 0} \frac{1}{2}(\sigma - \hat{\Sigma}_{n,n,i}^{(k)})^2 + \tau\frac{\gamma e^\gamma}{(\gamma + \hat{S}_{n,n,i}^{(k,l)})^2}|\sigma|. \tag{3.17}$$

According to the soft-thresholding operator [9], (3.17) has a closed-form solution

$$\hat{S}^{(k,l+1)}_{n,n,i} = \left(\hat{\Sigma}^{(k)}_{n,n,i} - \tau \frac{\gamma e^{\gamma}}{(\gamma + \hat{S}^{(k,l)}_{n,n,i})^2} \right)_+ .$$

Let

$$g(\hat{S}_{n,n,i}) = \left(\hat{\Sigma}^{(k)}_{n,n,i} - \tau \frac{\gamma e^{\gamma}}{(\gamma + \hat{S}_{n,n,i})^2} \right)_+ , \quad 0 \le \hat{S}_{n,n,i} < \hat{\Sigma}^{(k)}_{n,n,i}.$$

Since $\phi(\sigma)$ is concave in σ, at each iteration its value decreases by an amount more than the decrease in the value of the linearized objective function. Following Lemma 3, the iteration $\hat{S}^{(k,l+1)}_{n,n,i} = g(\hat{S}^{(k,l)}_{n,n,i})$ converges to the local minimum $\hat{S}^{(k)}_{n,n,i}$ after a number of iterations. We thus complete the proof. □

By summarizing the aforementioned solving process, we show the pseudocode of the developed ADMM algorithm for solving Model (3.6) in Algorithm 2.

Algorithm 2 Solve the non-convex TRPCA Model (3.6) by ADMM

```
Input: The observed data $\mathcal{X} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$; balanced parameter $\lambda$; error tolerance $tol$; maximum iteration $K$; parameters $p$, $\gamma$, and $\epsilon$.
    Initialize: $\mathcal{L}_0 = \mathcal{E}_0 = \mathcal{M}_0 = 0$, $\beta_0 = 10^{-4}$, $\beta_{max} = 10^{10}$, and $\eta = 1.1$.
while not converged and $k < K$ do
    $\mathcal{E}_{k+1} \leftarrow \mathrm{GST}\left(\mathcal{B}_k, \frac{\lambda \mathcal{W}_k^p}{\beta_k}, p\right)$ with $\mathcal{B}_k = \mathcal{X} - \mathcal{L}_k - \frac{\mathcal{M}_k}{\beta_k}$ and $\left(\mathcal{W}_k^p\right)_{ijk} = \left(\frac{1}{|\mathcal{E}^{(k)}_{ijk}| + \epsilon}\right)^p$;
    $\mathcal{L}_{k+1} \leftarrow \arg\min_{\mathcal{L}} \frac{1}{2}\|\mathcal{A}_k - \mathcal{L}\|_F^2 + \tau\|\mathcal{L}\|_{t-e^{\gamma}}$ with $\mathcal{A}_k = \mathcal{X} - \mathcal{E}_{k+1} - \frac{\mathcal{M}_k}{\beta_k}$ and $\tau = \frac{1}{\beta_k}$;
    $\mathcal{M}_{k+1} \leftarrow \mathcal{M}_k + \beta_k(\mathcal{L}_{k+1} + \mathcal{E}_{k+1} - \mathcal{X})$;
    $\beta_{k+1} \leftarrow \min\{\eta\beta_k, \beta_{max}\}$;
    Check the convergence conditions:
        $\max\left\{\|\mathcal{L}_{k+1} - \mathcal{L}_k\|_\infty, \|\mathcal{E}_{k+1} - \mathcal{E}_k\|_\infty, \|\mathcal{L}_{k+1} + \mathcal{E}_{k+1} - \mathcal{X}\|_\infty\right\} \le tol;$    (3.18)
    $k = k + 1$.
end while
Output: The low rank tensor $\mathcal{L}$ and the sparse tensor $\mathcal{E}$.
```

3.3 Computational Complexity

Suppose that after finite iterations, we can get optimal solutions of the fixed point iterations (3.10) and (3.13), respectively. Then the computation cost of the proposed algorithm mainly lies in updating $\mathcal{L}$. To update $\mathcal{L}$, in each iteration, we need to compute the FFT along the third mode, calculate singular values of all frontal slices by using SVD, and compute the inverse FFT along the third mode.

As we all know, for an $M_1 \times M_2$ matrix, the computational complexity of the SVD is $O(M_1 M_2 \min\{M_1, M_2\})$; the computational complexity of the FFT on an M-dimension vector is $O(M \log(M))$.

Given an observed tensor $\mathcal{X} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$. To update $\mathcal{L} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, we first calculate $\hat{\mathcal{L}} = \text{fft}(\mathcal{L}, [\ ], 3)$ along the third mode and the computational complexity of this process is $O(N_1 N_2 N_3 \log(N_3))$; then we compute the SVD of matrices $\hat{\mathbf{L}}_i \in \mathbb{R}^{N_1 \times N_2} (i = 1, \ldots, N_3)$ in the Fourier domain and it will take $O(N_1 N_2 N_3 \min\{N_1, N_2\})$; finally, calculating $\mathcal{L} = \text{ifft}(\hat{\mathcal{L}}, [\ ], 3)$ along the third mode will take $O(N_1 N_2 N_3 \log(N_3))$. In summary, in each iteration, the total computational complexity of the proposed Algorithm is $O(2N_1 N_2 N_3 \log(N_3) + N_1 N_2 N_3 \min\{N_1, N_2\})$. For simplicity, if $N_1 \approx N_2 \approx N_3 = N$, the cost in each iteration is $O(N^4)$.

4 Convergence Analysis of the Proposed Algorithm

In this section, we provide a theoretical guarantee for the convergence of the developed ADMM Algorithm 2. Before giving the convergence analysis, we first show that the sequences $\{\mathcal{L}_k\}$, $\{\mathcal{E}_k\}$, $\{\mathcal{M}_k\}$ generated by Algorithm 2 are bounded.

Lemma 4 *The sequence $\{\mathcal{M}_k\}$ generated by Algorithm 2 is bounded.*

Proof According to (3.8), at the $(k+1)$-th iteration, we have

$$\begin{aligned}
\|\mathcal{M}_{k+1}\|_F^2 &= \|\mathcal{M}_k + \beta_k \left(\mathcal{L}_{k+1} + \mathcal{E}_{k+1} - \mathcal{X}\right)\|_F^2 \\
&= \beta_k^2 \|\frac{1}{\beta_k}\mathcal{M}_k + \mathcal{L}_{k+1} + \mathcal{E}_{k+1} - \mathcal{X}\|_F^2 \\
&= \beta_k^2 \|\mathcal{A}_k - \mathcal{L}_{k+1}\|_F^2 \\
&= \frac{\beta_k^2}{N_3} \sum_{i=1}^{N_3} \|\hat{\mathbf{A}}_{k,i} - \hat{\mathbf{L}}_{k+1,i}\|_F^2 \\
&= \frac{\beta_k^2}{N_3} \sum_{i=1}^{N_3} \|\hat{\mathbf{U}}_{k,i} \left(\hat{\mathbf{\Sigma}}_{k,i} - \hat{\mathbf{S}}_{k,i}\right) \hat{\mathbf{V}}_{k,i}^T\|_F^2 \\
&= \frac{\beta_k^2}{N_3} \sum_{i=1}^{N_3} \|\hat{\mathbf{\Sigma}}_{k,i} - \hat{\mathbf{S}}_{k,i}\|_F^2 \\
&= \frac{\beta_k^2}{N_3} \sum_{i=1}^{N_3} \sum_{n=1}^{\min\{N_1,N_2\}} \left(\hat{\Sigma}_{n,n,i}^{(k)} - \hat{S}_{n,n,i}^{(k)}\right)^2,
\end{aligned} \tag{4.1}$$

where the fourth equality holds due to Remark 3, the fifth one follows from Theorem 1, the sixth one holds due to the unitary invariance of the Frobenius norm.

Recall (3.16) in Theorem 1

$$\hat{S}_{n,n,i}^{(k)} = \arg\min_{\sigma \geq 0} \frac{1}{2}(\sigma - \hat{\Sigma}_{n,n,i}^{(k)})^2 + \tau \frac{e^{\gamma}\sigma}{\gamma + \sigma},$$

we thus get the following KKT conditions:

$$\begin{cases} \left(\hat{S}^{(k)}_{n,n,i} - \hat{\Sigma}^{(k)}_{n,n,i}\right) + \tau \dfrac{e^{\gamma}\gamma}{\left(\gamma + \hat{S}^{(k)}_{n,n,i}\right)^2} - \mu = 0, \\ \mu \cdot \hat{S}^{(k)}_{n,n,i} = 0, \\ \hat{S}^{(k)}_{n,n,i} \geq 0, \\ \mu \geq 0, \end{cases} \tag{4.2}$$

where μ is the Lagrange multiplier.

If $\mu = \hat{S}^{(k)}_{n,n,i} = 0$, we have

$$\left(\hat{S}^{(k)}_{n,n,i} - \hat{\Sigma}^{(k)}_{n,n,i}\right)^2 = \left(\hat{\Sigma}^{(k)}_{n,n,i}\right)^2 = \left(\tau \frac{e^{\gamma}\gamma}{\left(\gamma + \hat{S}^{(k)}_{n,n,i}\right)^2}\right)^2 \leq \frac{(\tau e^{\gamma})^2}{\gamma^2}.$$

If $\mu = 0$, $\hat{S}^{(k)}_{n,n,i} \neq 0$, we have

$$\left(\hat{S}^{(k)}_{n,n,i} - \hat{\Sigma}^{(k)}_{n,n,i}\right)^2 = \left(\tau \frac{e^{\gamma}\gamma}{\left(\gamma + \hat{S}^{(k)}_{n,n,i}\right)^2}\right)^2 \leq \frac{(\tau e^{\gamma})^2}{\gamma^2}.$$

If $\mu \neq 0$, $\hat{S}^{(k)}_{n,n,i} = 0$, then $\left(-\hat{\Sigma}^{(k)}_{n,n,i}\right) + \tau \dfrac{e^{\gamma}}{\gamma} - \mu = 0$. This means $\hat{\Sigma}^{(k)}_{n,n,i} = \tau \dfrac{e^{\gamma}}{\gamma} - \mu \leq \dfrac{\tau e^{\gamma}}{\gamma}$ due to γ, τ, μ are positive numbers. Noting that $\hat{\Sigma}^{(k)}_{n,n,i}$ is the n-th singular value of $\hat{\mathbf{A}}_{k,i}$, then $0 \leq \hat{\Sigma}^{(k)}_{n,n,i} \leq \dfrac{\tau e^{\gamma}}{\gamma}$ holds for all $i = 1, \ldots, N_3$ and $n = 1, \ldots, \min\{N_1, N_2\}$. That is to say, in this case, $\left(\hat{\Sigma}^{(k)}_{n,n,i}\right)^2$ has an upper bound $\dfrac{(\tau e^{\gamma})^2}{\gamma^2}$.

As discussed above, we always have $\left(\hat{\Sigma}^{(k)}_{n,n,i} - \hat{S}^{(k)}_{n,n,i}\right)^2 \leq \dfrac{(\tau e^{\gamma})^2}{\gamma^2}$. This implies that (4.1) has an upper bound

$$\frac{\beta_k^2}{N_3} \sum_{i=1}^{N_3} \sum_{n=1}^{\min\{N_1,N_2\}} \frac{(\tau e^{\gamma})^2}{\gamma^2} = \min\{N_1, N_2\} \cdot \frac{e^{2\gamma}}{\gamma^2},$$

which proves the lemma. □

Lemma 5 *Both sequences $\{\mathcal{L}_k\}$ and $\{\mathcal{E}_k\}$ generated by Algorithm 2 are bounded.*

Proof According to (3.7) and Step 3 in (3.8), we have

$$\begin{aligned} &L_{\beta_k}(\mathcal{L}_k, \mathcal{E}_k, \mathcal{M}_k) \\ &\quad = L_{\beta_{k-1}}(\mathcal{L}_k, \mathcal{E}_k, \mathcal{M}_{k-1}) \\ &\qquad + \frac{\beta_k - \beta_{k-1}}{2} \|\mathcal{L}_k + \mathcal{E}_k - \mathcal{X}\|_F^2 + \beta_{k-1} \|\mathcal{X} - \mathcal{L}_k - \mathcal{E}_k\|_F^2 \\ &\quad = L_{\beta_{k-1}}(\mathcal{L}_k, \mathcal{E}_k, \mathcal{M}_{k-1}) + \frac{\beta_k + \beta_{k-1}}{2(\beta_{k-1})^2} \|\mathcal{M}_k - \mathcal{M}_{k-1}\|_F^2. \end{aligned} \tag{4.3}$$

Since

$$\mathcal{E}_{k+1} = \arg\min_{\mathcal{E}} L_{\beta_k}\left(\mathcal{L}_k, \mathcal{E}, \mathcal{M}_k\right),$$
$$\mathcal{L}_{k+1} = \arg\min_{\mathcal{L}} L_{\beta_k}\left(\mathcal{L}, \mathcal{E}_{k+1}, \mathcal{M}_k\right),$$

we have

$$\begin{aligned} L_{\beta_k}\left(\mathcal{L}_{k+1}, \mathcal{E}_{k+1}, \mathcal{M}_k\right) &\le L_{\beta_k}\left(\mathcal{L}_k, \mathcal{E}_{k+1}, \mathcal{M}_k\right) \le L_{\beta_k}\left(\mathcal{L}_k, \mathcal{E}_k, \mathcal{M}_k\right) \\ &= L_{\beta_{k-1}}\left(\mathcal{L}_k, \mathcal{E}_k, \mathcal{M}_{k-1}\right) + \frac{\beta_k + \beta_{k-1}}{2\left(\beta_{k-1}\right)^2}\|\mathcal{M}_k - \mathcal{M}_{k-1}\|_F^2. \end{aligned} \tag{4.4}$$

Iterating (4.4) k times, we can obtain

$$\begin{aligned} L_{\beta_k}\left(\mathcal{L}_{k+1}, \mathcal{E}_{k+1}, \mathcal{M}_k\right) &\le L_{\beta_0}\left(\mathcal{L}_1, \mathcal{E}_1, \mathcal{M}_0\right) + \sum_{i=1}^{k} \frac{\beta_i + \beta_{i-1}}{2\left(\beta_{i-1}\right)^2}\|\mathcal{M}_i - \mathcal{M}_{i-1}\|_F^2 \\ &\le L_{\beta_0}\left(\mathcal{L}_0, \mathcal{E}_1, \mathcal{M}_0\right) + \sum_{i=1}^{k} \frac{\beta_i + \beta_{i-1}}{2\left(\beta_{i-1}\right)^2}\|\mathcal{M}_i - \mathcal{M}_{i-1}\|_F^2 \\ &\le L_{\beta_0}\left(\mathcal{L}_0, \mathcal{E}_0, \mathcal{M}_0\right) + \sum_{i=1}^{k} \frac{\beta_i + \beta_{i-1}}{2\left(\beta_{i-1}\right)^2}\|\mathcal{M}_i - \mathcal{M}_{i-1}\|_F^2 \qquad (4.5) \\ &= \frac{\beta_0}{2}\|\mathcal{X}\|_F^2 + \sum_{i=1}^{k} \frac{\beta_i + \beta_{i-1}}{2\left(\beta_{i-1}\right)^2}\|\mathcal{M}_i - \mathcal{M}_{i-1}\|_F^2 \\ &\le \frac{\beta_0}{2}\|\mathcal{X}\|_F^2 + \left(\max_i \|\mathcal{M}_i - \mathcal{M}_{i-1}\|_F^2\right) \cdot \sum_{i=1}^{k} \frac{\beta_i + \beta_{i-1}}{2\left(\beta_{i-1}\right)^2}. \end{aligned}$$

Since $\{\mathcal{M}_k\}$ is bounded, it follows that the quantity $\max_i \|\mathcal{M}_i - \mathcal{M}_{i-1}\|_F^2$ is also bounded. Notice that $\beta_i = \eta\beta_{i-1} = \eta^i\beta_0$, $\eta = 1.1$, $\beta_0 = 10^{-4}$, then

$$\sum_{i=1}^{\infty} \frac{\beta_i + \beta_{i-1}}{2\left(\beta_{i-1}\right)^2} = \frac{\eta + 1}{2\beta_0} \sum_{i=1}^{\infty} \frac{1}{\eta^{i-1}} = \frac{\eta(\eta+1)}{2\beta_0(\eta-1)}$$

is bounded, and hence $L_{\beta_k}\left(\mathcal{L}_{k+1}, \mathcal{E}_{k+1}, \mathcal{M}_k\right)$ has upper bound.

On the other hand, we have

$$\begin{aligned} & L_{\beta_k}\left(\mathcal{L}_{k+1}, \mathcal{E}_{k+1}, \mathcal{M}_k\right) + \frac{1}{2\beta_k}\|\mathcal{M}_k\|_F^2 \\ &= \|\mathcal{L}_{k+1}\|_{t-e^{\gamma}} + \lambda\|\mathcal{E}_{k+1}\|_{\mathcal{W}_{k+1},\ell_p}^p + \frac{\beta_k}{2}\|\mathcal{L}_{k+1} - \mathcal{X} + \mathcal{E}_{k+1} + \frac{\mathcal{M}_k}{\beta_k}\|_F^2, \end{aligned} \tag{4.6}$$

and noting that each term on the right-hand side of the equation (4.6) is nonnegative, we thus obtain the sequences $\{\mathcal{L}_{k+1}\}$ and $\{\mathcal{E}_{k+1}\}$ are bounded. □

Lemma 6 [23] *Suppose $F : \mathbb{R}^{N_1 \times N_2} \to \mathbb{R}$ is represented as $F(\mathbf{X}) = f \circ \sigma(\mathbf{X})$, where $\mathbf{X} \in \mathbb{R}^{N_1 \times N_2}$ with the SVD is $\mathbf{X} = \mathbf{U}\,\mathrm{diag}(\sigma(\mathbf{X}))\mathbf{V}^T$, and f is differentiable. Then the gradient of $F(\mathbf{X})$ at $\mathbf{X}$ is*

$$\frac{\partial F(\mathbf{X})}{\partial \mathbf{X}} = \mathbf{U}\,\mathrm{diag}(\theta)\mathbf{V}^T,$$

where $\theta = \frac{\partial f(y)}{\partial y}|_{y=\sigma(\mathbf{X})}$.

Lemma 7 [32] *For the ℓ_p regularized unconstrained nonlinear programming model*

$$\min_{\mathbf{x}\in\mathbb{R}^n}\left\{F(\mathbf{x}) := f(\mathbf{x}) + \lambda\|\mathbf{x}\|_p^p\right\}, \tag{4.7}$$

where $\lambda > 0$, $p \in (0,1)$, $\|\mathbf{x}\|_p := \left(\sum_{i=1}^n |x_i|^p\right)^{1/p}$, f is a smooth function with L_f-Lipschitz continuous gradient in $\mathbb{R}^n$, that is

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|_2 \le L_f\|\mathbf{x}-\mathbf{y}\|_2, \quad \forall \mathbf{x},\mathbf{y}\in\mathbb{R}^n,$$

and f is bounded below in $\mathbb{R}^n$. Then the following statements and results hold:

(i) Let $\mathbf{x}^*$ be a local minimizer of (4.7) and $\mathbf{X}^* = \operatorname{diag}(\mathbf{x}^*)$. Then $\mathbf{x}^*$ is a first-order stationary point, that is, $\mathbf{X}^*\nabla f(\mathbf{x}^*) + \lambda p|\mathbf{x}^*|^p = 0$ holds at $\mathbf{x}^*$, where $|\mathbf{x}^*|^p = (|x_1^*|^p, \cdots, |x_n^*|^p)^T$ is an n dimensional vector.

(ii) Let $\mathbf{x}^*$ be a first-order stationary point of (4.7) satisfying $F(\mathbf{x}^*) \le F\left(\mathbf{x}^0\right) + \epsilon_1$ for some $\mathbf{x}^0\in\mathbb{R}^n$ and a tiny $\epsilon_1 > 0$. Defining $\underline{f} = \inf_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x})$ and $\operatorname{supp}(\mathbf{x}^*) = \left\{i : x_i^* \ne 0\right\}$, then

$$\left|x_i^*\right| \ge \left(\frac{\lambda p}{\sqrt{2L_f\left[F\left(\mathbf{x}^0\right)+\epsilon_1-\underline{f}\right]}}\right)^{\frac{1}{1-p}}, \quad \forall i \in \operatorname{supp}(\mathbf{x}^*).$$

With the help of the preceding results, we can now present the convergence analysis of Algorithm 2.

Theorem 2 *Let the sequence $\mathcal{P}_k = \{\mathcal{L}_k, \mathcal{E}_k, \mathcal{M}_k\}$ be generated by Algorithm 2. Then the accumulation point $\mathcal{P}^* = \{\mathcal{L}^*, \mathcal{E}^*, \mathcal{M}^*\}$ of $\mathcal{P}_k$ is a KKT stationary point, i.e., $\mathcal{P}^*$ satisfies the following KKT conditions:*

$$0 \in \partial\|\mathcal{L}^*\|_{t-e^\gamma} + \mathcal{M}^*, \quad \lambda\cdot p\cdot\|\mathcal{E}^*\|_{\mathcal{W},\ell_p}^p + \left\langle\mathcal{E}^*, \mathcal{M}^*\right\rangle = 0, \quad \mathcal{L}^* + \mathcal{E}^* = \mathcal{X}.$$

Proof From Lemmas 4 and 5, the sequence $\mathcal{P}_k = \{\mathcal{L}_k, \mathcal{E}_k, \mathcal{M}_k\}$ generated by Algorithm 2 is bounded. According to the Bolzano-Weierstrass theorem, the sequence $\{\mathcal{P}_k\}$ has at least one convergent subsequence, thus there must be at least one accumulation point $\mathcal{P}^* = \{\mathcal{L}^*, \mathcal{E}^*, \mathcal{M}^*\}$ for the sequence $\mathcal{P}_k$. Without loss of generality, we assume that $\{\mathcal{P}_k\}$ converges to $\mathcal{P}^*$.

By (3.8), we have

$$\left(\mathcal{M}_{k+1} - \mathcal{M}_k\right)/\beta_k = \mathcal{L}_{k+1} + \mathcal{E}_{k+1} - \mathcal{X}. \tag{4.8}$$

Taking the limit on both sides of the above formula gives:

$$\lim_{k\to\infty}\left(\mathcal{L}_{k+1} + \mathcal{E}_{k+1} - \mathcal{X}\right) = \lim_{k\to\infty}\left(\mathcal{M}_{k+1} - \mathcal{M}_k\right)/\beta_k = 0, \tag{4.9}$$

and hence we have

$$\mathcal{L}^* + \mathcal{E}^* = \mathcal{X}. \tag{4.10}$$

By

$$\mathcal{L}_{k+1} = \arg\min_{\mathcal{L}} L_{\beta_k}\left(\mathcal{L}, \mathcal{E}_{k+1}, \mathcal{M}_k\right),$$

we have

$$\begin{aligned} 0 &\in \partial\|\mathcal{L}\|_{t-e^\gamma}|_{\mathcal{L}_{k+1}} + \mathcal{M}_k + \beta_k\left(\mathcal{L}_{k+1} + \mathcal{E}_{k+1} - \mathcal{X}\right) \\ &= \partial\|\mathcal{L}\|_{t-e^\gamma}|_{\mathcal{L}_{k+1}} + \mathcal{M}_{k+1}. \end{aligned} \tag{4.11}$$

From Lemma 6, we know

$$\frac{\partial \|\hat{\mathbf{L}}_{n_3}\|_{e^\gamma}}{\partial \hat{\mathbf{L}}_{n_3}} = \hat{\mathbf{U}}_{\mathcal{L},n_3} \operatorname{diag}\left(\frac{\gamma e^\gamma}{\left(\gamma + \sigma_1(\hat{\mathbf{L}}_{n_3})\right)^2}, \cdots, \frac{\gamma e^\gamma}{\left(\gamma + \sigma_n(\hat{\mathbf{L}}_{n_3})\right)^2}\right)^T \hat{\mathbf{V}}_{\mathcal{L},n_3}^T$$

for all $n_3 = 1, \ldots, N_3$, where $n = \min\{N_1, N_2\}$. Noting that $\frac{\gamma e^\gamma}{\left(\gamma + \sigma_i(\hat{\mathbf{L}}_{n_3})\right)^2} \le \frac{e^\gamma}{\gamma}$ holds for the given $\gamma > 0$ and any $\sigma_i(\hat{\mathbf{L}}_{n_3})$ $(i = 1, \ldots, n)$. Thus

$$\|\frac{\partial \|\hat{\mathbf{L}}_{n_3}\|_{e^\gamma}}{\partial \hat{\mathbf{L}}_{n_3}}\|_F^2 \le n \cdot \frac{e^\gamma}{\gamma}, \quad n_3 = 1, \ldots, N_3.$$

Therefore, it is easily to see that

$$\frac{\partial \|\mathcal{L}\|_{t-e^\gamma}}{\partial \hat{\mathcal{L}}} = \left[\frac{\partial \|\hat{\mathbf{L}}_1\|_{e^\gamma}}{\partial \hat{\mathbf{L}}_1} | \cdots | \frac{\partial \|\hat{\mathbf{L}}_{N_3}\|_{e^\gamma}}{\partial \hat{\mathbf{L}}_{N_3}}\right]$$

is bounded. Using $\hat{\mathcal{L}} = \mathcal{L} \times_3 \widetilde{\mathbf{F}}_{N_3}$ and the chain rule [7] give

$$\|\frac{\partial \|\mathcal{L}\|_{t-e^\gamma}}{\partial \mathcal{L}}\|_F^2 = \|\frac{\partial \|\mathcal{L}\|_{t-e^\gamma}}{\partial \hat{\mathcal{L}}} \times_3 \widetilde{\mathbf{F}}_{N_3}^* \|_F^2 \le n \cdot N_3 \cdot \frac{e^\gamma}{\gamma}.$$

From (4.11), let $k \to \infty$, we have

$$0 \in \partial \|\mathcal{L}^*\|_{t-e^\gamma} + \mathcal{M}^*. \tag{4.12}$$

Similarly, since

$$\mathcal{E}_{k+1} = \arg\min_{\mathcal{E}} L_{\beta_k}(\mathcal{L}_k, \mathcal{E}, \mathcal{M}_k),$$

according to Lemma 7, we have the following first-order optimality condition

$$\lambda \cdot p \cdot \|\mathcal{E}_{k+1}\|_{\mathcal{W},\ell_p}^p + \langle \mathcal{E}_{k+1}, \mathcal{M}_{k+1} - \beta_k(\mathcal{L}_{k+1} - \mathcal{L}_k)\rangle = 0. \tag{4.13}$$

Noting that $\beta_k < \beta_{max} = 10^{10}$, then when $k \to \infty$, we have $\beta_k(\mathcal{L}_{k+1} - \mathcal{L}_k) \to 0$. By (4.13), we have

$$\lambda \cdot p \cdot \|\mathcal{E}^*\|_{\mathcal{W},\ell_p}^p + \left\langle \mathcal{E}^*, \mathcal{M}^* \right\rangle = 0,$$

which together with (4.10) and (4.12) give the assertion. □

Theorem 3 *The series $\{\mathcal{L}_k\}$ and $\{\mathcal{E}_k\}$ generated by Algorithm 2 are Cauchy sequences, and converge to some critical points of $L_\beta(\mathcal{L}, \mathcal{E}, \mathcal{M})$.*

Proof We first show that $\{\mathcal{E}_k\}$ is a Cauchy sequence. From the third equation in (3.8), we have

$$\mathcal{E}_t = \mathcal{X} - \mathcal{L}_t + \frac{1}{\beta_{t-1}}(\mathcal{M}_t - \mathcal{M}_{t-1}).$$

Thus,

$$\begin{aligned}&\|\mathcal{E}_{t+1}-\mathcal{E}_t\|_F\\&=\|\mathcal{E}_{t+1}-\left(\mathcal{X}-\mathcal{L}_t-\frac{1}{\beta_t}\mathcal{M}_t\right)-\frac{1}{\beta_t}\mathcal{M}_t-\frac{1}{\beta_{t-1}}\left(\mathcal{M}_t-\mathcal{M}_{t-1}\right)\|_F\\&\le\|\mathcal{E}_{t+1}-\mathcal{B}_t\|_F+\|\frac{1}{\beta_t}\mathcal{M}_t+\frac{1}{\beta_{t-1}}\left(\mathcal{M}_t-\mathcal{M}_{t-1}\right)\|_F.\end{aligned}\tag{4.14}$$

By (3.8), we have

$$\mathcal{E}_{ijk}^{(t+1)}=\arg\min_{\mathcal{E}_{ijk}}\lambda\left(\mathcal{W}_{ijk}^{(t)}|\mathcal{E}_{ijk}|\right)^p+\frac{\beta_t}{2}\left(b_{ijk}^{(t)}-\mathcal{E}_{ijk}\right)^2,\tag{4.15}$$

it follows from Lemma 7 that

$$\lambda\cdot p\cdot\left(\mathcal{W}_{ijk}^{(t)}|\mathcal{E}_{ijk}^{(t+1)}|\right)^p+\beta_t\cdot\mathcal{E}_{ijk}^{(t+1)}\cdot\left(\mathcal{E}_{ijk}^{(t+1)}-b_{ijk}^{(t)}\right)=0.\tag{4.16}$$

From (4.15) and (4.16), it is easy to see that $\mathcal{E}_{ijk}^{(t+1)}=0$ if and only if $b_{ijk}^{(t)}=0$.

If $\mathcal{E}_{ijk}^{(t+1)}\neq 0$, we have

$$\left(\mathcal{E}_{ijk}^{(t+1)}-b_{ijk}^{(t)}\right)^2=\left(\frac{\lambda\cdot p\cdot\left(\mathcal{W}_{ijk}^{(t)}|\mathcal{E}_{ijk}^{(t+1)}|\right)^p}{\beta_t\cdot\mathcal{E}_{ijk}^{(t+1)}}\right)^2\le\left(\frac{\lambda\cdot\left(\mathcal{W}_{ijk}^{(t)}\right)^p\cdot|\mathcal{E}_{ijk}^{(t+1)}|^{p-1}}{\beta_t}\right)^2$$

since $0<p<1$. Noting that $\mathcal{W}_{ijk}^{(t)}=\dfrac{1}{|\mathcal{E}_{ijk}^{(t)}|+\epsilon}$, ϵ is a given constant. Thus we have

$$\left(\mathcal{W}_{ijk}^{(t)}\right)^{2p}\le\left(\frac{1}{\epsilon}\right)^{2p}.\tag{4.17}$$

On the other hand, according to the second result of Lemma 7, $\mathcal{E}_{ijk}^{(t+1)}\neq 0$ has a lower bound, which is denoted by δ. Thus we can obtain the following inequality

$$\left(|\mathcal{E}_{ijk}^{(t+1)}|\right)^{2p-2}\le(\delta)^{2p-2}.\tag{4.18}$$

Denoting the upper bounds in inequalities (4.17) and (4.18) by M_1 and M_2, respectively. Therefore,

$$\left(\mathcal{E}_{ijk}^{(t+1)}-b_{ijk}^{(t)}\right)^2\le\frac{\lambda^2\cdot M_1\cdot M_2}{\beta_t^2}\quad\text{if}\quad\mathcal{E}_{ijk}^{(t+1)}\neq 0.$$

If $\mathcal{E}_{ijk}^{(t+1)}=0$, we also have $b_{ijk}^{(t)}=0$. And thus we have

$$\left(\mathcal{E}_{ijk}^{(t+1)}-b_{ijk}^{(t)}\right)^2=0\le\frac{\lambda^2\cdot M_1\cdot M_2}{\beta_t^2}.$$

In conclude, we have

$$\begin{aligned}\|\mathcal{E}_{t+1}-\mathcal{E}_t\|_F\le&\frac{\lambda\cdot\sqrt{(N_1N_2N_3)\cdot M_1\cdot M_2}}{\beta_t}+\frac{1}{\beta_t}\|\mathcal{M}_t\|_F\\&+\frac{1}{\beta_{t-1}}\|\mathcal{M}_t\|_F+\frac{1}{\beta_{t-1}}\|\mathcal{M}_{t-1}\|_F,\end{aligned}$$

where $N_1 \times N_2 \times N_3$ is the size of the tensor $\mathcal{X}$. Noting that $\{\mathcal{M}_t\}$ is a bounded sequence and $\beta_{t+1} = \eta\beta_t$, $\eta = 1.1$, $\beta_0 = 10^{-4}$, we thus have

$$\lim_{m,n\to\infty} \|\mathcal{E}_m - \mathcal{E}_n\|_F \le \lim_{m,n\to\infty} \sum_{t=n}^{m} \|\mathcal{E}_{t+1} - \mathcal{E}_t\|_F = 0$$

for any $n < m$. We thus proved $\{\mathcal{E}_t\}$ is a Cauchy sequence.

Next, we prove that $\{\mathcal{L}_t\}$ is a Cauchy sequence. From the third equation in (3.8), we have

$$\mathcal{L}_{t+1} = \mathcal{X} - \mathcal{E}_{t+1} + \frac{1}{\beta_t}\left(\mathcal{M}_{t+1} - \mathcal{M}_t\right).$$

Thus,

$$\begin{aligned}
&\|\mathcal{L}_{t+1} - \mathcal{L}_t\|_F \\
&\quad = \|\mathcal{X} - \mathcal{E}_{t+1} + \frac{1}{\beta_t}\left(\mathcal{M}_{t+1} - \mathcal{M}_t\right) - \mathcal{L}_t + \left(\mathcal{E}_t + \frac{1}{\beta_{t-1}}\mathcal{M}_{t-1}\right) \\
&\qquad - \left(\mathcal{E}_t + \frac{1}{\beta_{t-1}}\mathcal{M}_{t-1}\right)\|_F \\
&\quad = \|\left(\mathcal{X} - \mathcal{E}_t - \frac{1}{\beta_{t-1}}\mathcal{M}_{t-1}\right) - \mathcal{L}_t + \mathcal{E}_t - \mathcal{E}_{t+1} \\
&\qquad + \frac{1}{\beta_t}\left(\mathcal{M}_{t+1} - \mathcal{M}_t\right) + \frac{1}{\beta_{t-1}}\mathcal{M}_{t-1}\|_F \\
&\quad \le \|\mathcal{A}_{t-1} - \mathcal{L}_t\|_F + \|\mathcal{E}_t - \mathcal{E}_{t+1}\|_F \\
&\qquad + \frac{1}{\beta_t}\|\mathcal{M}_{t+1}\|_F + \frac{1}{\beta_t}\|\mathcal{M}_t\|_F + \frac{1}{\beta_{t-1}}\|\mathcal{M}_{t-1}\|_F.
\end{aligned}$$

Proceeding as in the proof of Lemma 4, we can easily obtain the following inequality

$$\|\mathcal{A}_{t-1} - \mathcal{L}_t\|_F^2 \le \frac{1}{\beta_t^2} \cdot \frac{\min\{N_1, N_2\} \cdot e^{2\gamma}}{\gamma^2}.$$

Since $\{\mathcal{E}_t\}$ is a Cauchy sequence, $\{\mathcal{M}_t\}$ is a bounded sequence, it yields that

$$\lim_{m,n\to\infty} \|\mathcal{L}_m - \mathcal{L}_n\|_F \le \lim_{m,n\to\infty} \sum_{t=n}^{m} \|\mathcal{L}_{t+1} - \mathcal{L}_t\|_F = 0$$

for any $n < m$. We thus proved $\{\mathcal{L}_k\}$ is a Cauchy sequence.

It follows from Theorem 2 that the limit points of sequences $\{\mathcal{L}_k\}$ and $\{\mathcal{E}_k\}$ converge to some critical points of $L_\beta\left(\mathcal{L}, \mathcal{E}, \mathcal{M}\right)$. This completes the proof. □

5 Numerical Experiments

In this section, to verify the efficiency of the proposed model, we will report the experiment results on two kinds of problems, i.e., image recovery from observations corrupted by random noise and background modeling for surveillance video, where image recovery including color image, brain magnetic resonance image (brain MRI) and gray video sequence. We compare the proposed model (3.6) (denoted by "t-e^{γ}-$\mathcal{W}$") with four existing models: the sum of the nuclear norms of all unfolding matrices of the tensor based model (denoted by "SNN") [25], the t-SVD based nuclear norm model (denoted by “TNN”) [29], the t-SVD based partial sum of the nuclear norm model (denoted by “PSTNN”) [16], the t-SVD based t-Gamma tensor quasi-norm model (denoted by “t-γ”) [2]. Besides, to show the effectiveness of the weighted ℓ_p-norm, we also compare the proposed model (3.6) with the following ℓ_p-norm based model

$$\min_{\mathcal{L},\mathcal{E}} \|\mathcal{L}\|_{t-e^\gamma} + \lambda\|\mathcal{E}\|_{\ell_p}^p, \quad \text{s.t.} \quad \mathcal{X} = \mathcal{L} + \mathcal{E}, \tag{5.1}$$

where $\|\mathcal{E}\|_{\ell_p}^p = \left(\sum_{i,j,k} |\mathcal{E}_{i,j,k}|^p\right)^{\frac{1}{p}}$ and $0 < p < 1$. For simplicity, we denote the model (5.1) by “t-e^γ”.

We first normalize the gray-scale value of the tested tensor to [0, 1]. The observed tensor is then obtained by randomly setting some pixels into random values in the range [0, 1]. All experiments are implemented in the Matlab R2019a Windows 10 environment and run on a desktop computer (AMD Ryzen 7-3750H, @ 2.3GHz, 16G RAM).

5.1 Parameter Setting

The ADMM has been widely used to solve the problems of tensor completion and TRPCA [2,11,16,26,27,41]. There are different choices for initialization parameters in different models. For the TNN, PSTNN, and t-γ models, parameters will be adjusted according to the authors’ suggestions [2,16,29]. For SNN, numerical experiments on TRPCA by using SNN was performed by Lu et al. [29]. Thus, parameter setting for the SNN will be adjusted according to [29]. For the proposed t-e^γ-$\mathcal{W}$ model (3.6) and the competing t-e^γ model (5.1), the parameters are tuned according to the parameter analysis given below.

The proposed t-e^γ-$\mathcal{W}$ model involves seven parameters: λ is the regularization parameter, which is used to balance the low rank and the sparse parts, respectively; β is the Lagrange penalty parameter; tol is the error tolerance; K is the maximum permission iterative number of times; γ is a positive parameter in the proposed non-convex rank approximation function; $0 < p < 1$ and $\epsilon > 0$ are two parameters in the weighted ℓ_p-norm. The t-e^γ model involves six parameters: λ, β, tol, K, γ, p. For these two models, parameters β, tol and K are set consistent with [2] for a fair comparison, i.e., β is initialized to 10^{-4} and iteratively increased by $\beta_{k+1} = \eta\beta_k$ with $\eta = 1.1$; tol and K are set as $tol = 10^{-8}$ and $K = 500$, respectively. For other parameters (λ, γ, p, ϵ for the t-e^γ-$\mathcal{W}$ model and λ, γ, p for the t-e^γ model), we illustrate their effect on image restoration by using two color images Bear and Horse (they are randomly selected from the BSD [33]).

- The regularization parameter λ has been studied in many literatures. For the TNN model (1.2), when the low rank and sparse components satisfy certain assumptions, regularization parameter is $\lambda_{TNN} = 1/\sqrt{\max(N_1, N_2)N_3}$ in theory to guarantee correct recovery (more details can be found in Theorem 4.1 of [29]). Therefore, λ_{TNN} is as a good rule of thumb, which can then be adjusted to obtain the best possible results in different models.
  - For the task of color image recovery in our experiments, the size of each color image is 481 × 321 × 3 or 321 × 481 × 3. According to [29], $\lambda_{TNN} = 0.0263$. Therefore, we search for the best value of λ nearby 0.0263 for the t-e^γ and t-e^γ-$\mathcal{W}$ models. The effect of the parameter λ on image restoration (Bear) with respect to PSNR, SSIM and RSE values (see Sect. 5.3 for the definitions) are shown in Fig. 2. We can see that the noise levels and the best parameter value of λ are in inverse proportion, i.e., the higher the noise level, the smaller λ is the better choice. Meanwhile, we can see that the best parameter value of λ in the t-e^γ-$\mathcal{W}$ model is smaller than that in

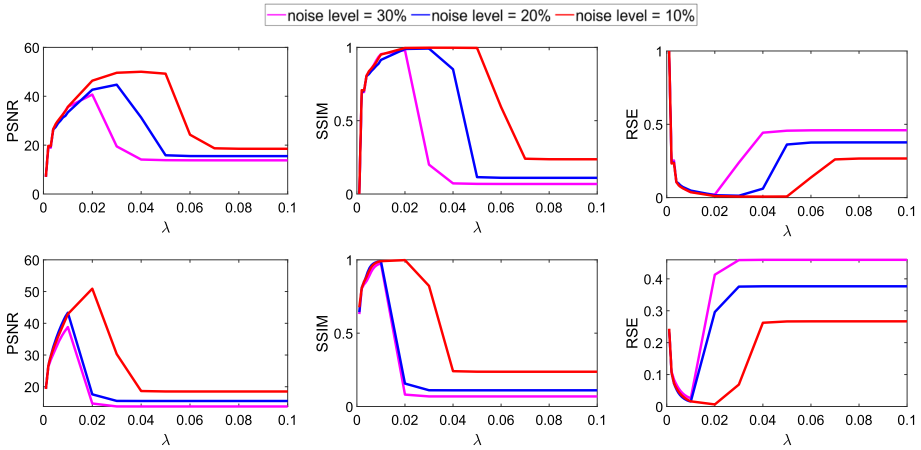


Fig. 2 The effect of the parameter λ on image restoration (Bear) with respect to PSNR, SSIM and RSE values. The first row is for the t-e^{γ} model, the second row is for the t-e^{γ}-$\mathcal{W}$ model

the t-e^{γ} model. Therefore, in the experiment of color image recovery, λ is selected from $[0.01, 0.06]$ and $[0.01, 0.03]$ with increment of 0.002 for the t-e^{γ} and t-e^{γ}-$\mathcal{W}$ models, respectively.

- When the third-dimension of the data is large (such as brain MRI data, video data, hyperspectral data), λ_{TNN} is usually very small. And thus a small λ would be better. In our experiments, both brain MRI and video data are medium sizes and $\lambda_{TNN} < 0.01$, thus the parameter λ is selected from $[0.001, 0.01]$ with increment of 0.001 for the t-e^{γ} and t-e^{γ}-$\mathcal{W}$ models when we conduct the experiments on brain MRI and videos.

- The effect of the parameter p on image restoration (Bear) with respect to PSNR, SSIM, RSE values are shown in Fig. 3. It can be seen from Fig. 3 that the best parameter value of p for t-e^{γ} and t-e^{γ}-$\mathcal{W}$ models are bigger than 0.5. Therefore, in the experiments, p is selected from $[0.5, 1]$ with increment of 0.1 for the two models.
- For different noise levels, the effect of the parameter γ on image restoration (Bear) with respect to PSNR are shown in Fig. 4. For the t-e^{γ} model, when $\gamma > 2$, the curves begin to decline. Therefore, $[0.1, 2]$ can be as a reference range for γ in the t-e^{γ} model. For the t-e^{γ}-$\mathcal{W}$ model, the curves reach their peaks at 1.5, 1.8 and 2.2, respectively. And after reaching the peaks, the curves begin to decline slowly. Therefore, $[1.5, 3]$ can be as a reference range for γ in the t-e^{γ}-$\mathcal{W}$ model. Therefore, in the experiments, γ is selected from $[0.1, 2]$ and $[1.5, 3]$ with increment of 0.1 for the t-e^{γ} and t-e^{γ}-$\mathcal{W}$ models, respectively.
- For the t-e^{γ}-$\mathcal{W}$ model, the influence of the parameter ϵ on image restoration (Bear and Horse) with respect to PSNR are plotted in Fig. 5. It can be observed that when $\epsilon = 0.1$, the curves get their peaks; when $\epsilon > 0.1$, the curves gradually tend to be stable. Therefore, in the experiments, we fixed ϵ to 0.1.

Since our experiments involve different applications associated with various datasets, we thus adjust the above parameters slightly to obtain the best performance results. For the sake of readability, we list all parameter settings for different models and different datasets in tables in the corresponding section.

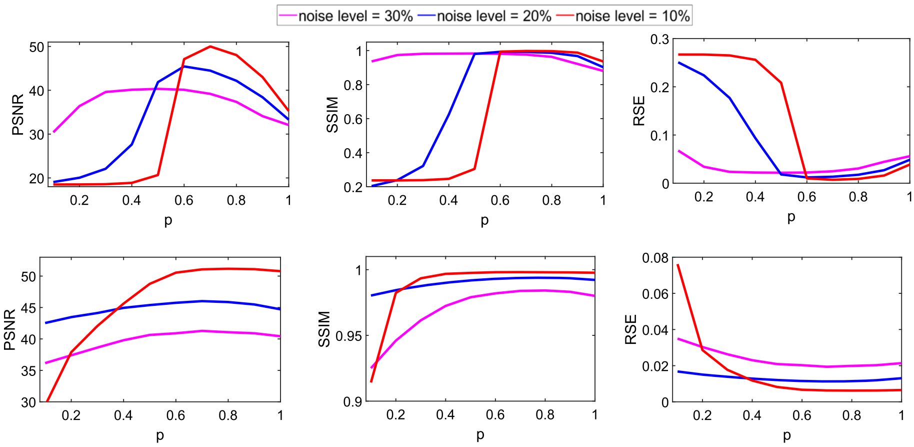


Fig. 3 The effect of the parameter p on image restoration (Bear) with respect to PSNR, SSIM and RSE values. The first row is for the t-e^{γ} model, the second row is for the t-e^{γ}-$\mathcal{W}$ model

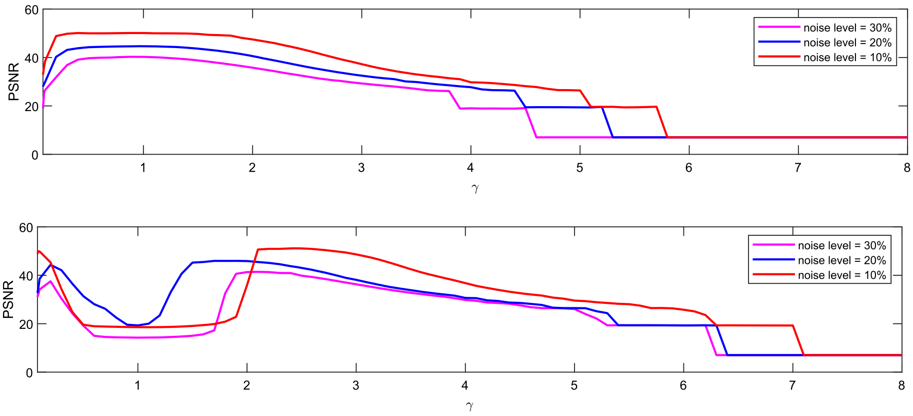


Fig. 4 The effect of the parameter γ on image restoration (Bear) with respect to PSNR values. The first row is for the t-e^{γ} model, the second row is for the t-e^{γ}-$\mathcal{W}$ model

5.2 Convergence Test

In this subsection, we perform the convergence test for the proposed t-e^{γ}-$\mathcal{W}$ model and the competing t-e^{γ} model with the application to color Tiger image recovery (the Tiger image can be seen in Fig. 9). Figure 6 shows the curve of PSNR versus outer iteration numbers, residual (given by (3.18)) versus outer iteration numbers, and RSE (given by (5.2)) versus outer iteration numbers, respectively. It can be observed that as the number of iterations increases, the curves gradually tend to be stable. This clearly demonstrates the convergence of the proposed t-e^{γ}-$\mathcal{W}$ and the competing t-e^{γ} models as shown in Theorem 3.

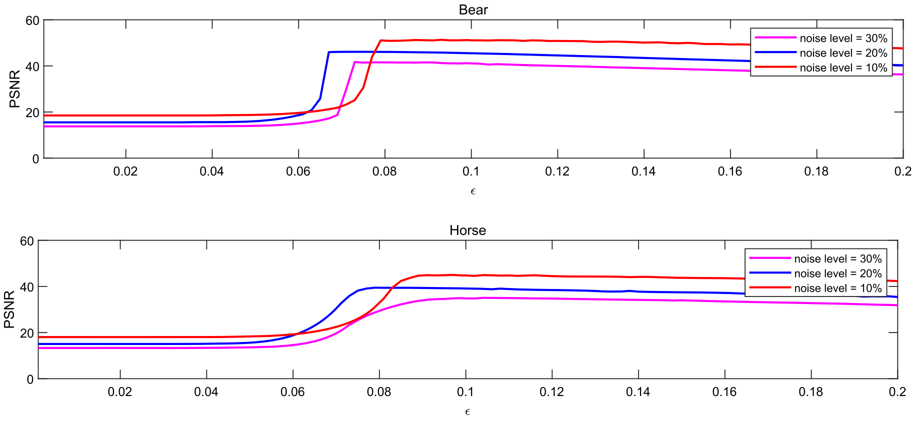


Fig. 5 The effect of the parameter ϵ on image restoration (Bear and Horse) with respect to PSNR values for the t-e^{γ}-$\mathcal{W}$ model

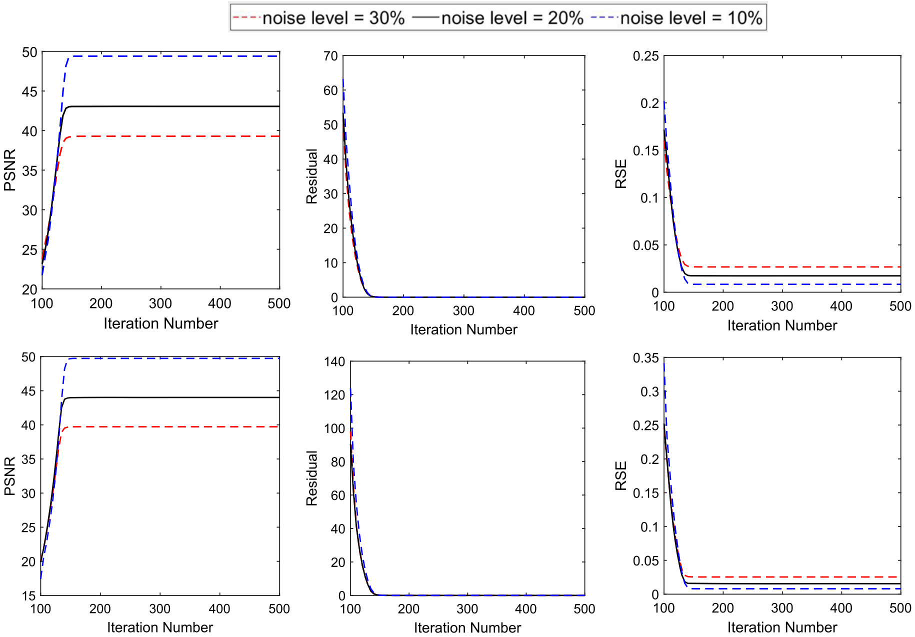


Fig. 6 Convergence test on color Tiger image with different noise levels by PSNR values, residual, and RSE, respectively. The first and second rows are the convergence test for the t-e^{γ} and t-e^{γ}-$\mathcal{W}$ models, respectively (Color figure online)

5.3 Image Recovery

In this subsection, we conduct numerical experiments on image recovery including color image, brain MRI and gray video sequence. The results are illustrated in Sects. 5.3.1, 5.3.2 and 5.3.3, respectively.

Some quantitative assessment indexes including the relative square error (RSE), peak signal-to-noise ratio (PSNR), structural similarity index (SSIM) are adopted to evaluate

Table 1 The parameter setting for color image recovery

Noise level	Model	p	γ	λ
30%, 20%, 10%	SNN	–	–	[15, 15, 1.5]
	TNN	–	–	$\lambda_{\text{TNN}} \approx 0.0263$
	PSTNN	–	–	$\lambda_{\text{TNN}} \approx 0.0263$
	t-γ	–	$\sqrt{\min\{N_1, N_2\}} \approx 17.9165$	$\lambda_{\text{TNN}} \approx 0.0263$
30%	t-e^{γ}	0.6	1.1	0.016
	t-e^{γ}-$\mathcal{W}$	0.9	2.3	0.014
20%	t-e^{γ}	0.7	1.2	0.026
	t-e^{γ}-$\mathcal{W}$	0.9	2.1	0.014
10%	t-e^{γ}	0.7	1.1	0.038
	t-e^{γ}-$\mathcal{W}$	0.9	2.4	0.024

the quality of the recovered images. The RSE and PSNR between the recovered tensor $\tilde{\mathcal{L}} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ and the original tensor $\mathcal{L} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ are defined as follows:

$$\text{RSE}(\tilde{\mathcal{L}}, \mathcal{L}) := \frac{\|\tilde{\mathcal{L}} - \mathcal{L}\|_F}{\|\mathcal{L}\|_F}, \tag{5.2}$$

$$\text{PSNR}(\tilde{\mathcal{L}}, \mathcal{L}) := 10 \log_{10}\left(\frac{N_1 N_2 N_3 \|\mathcal{L}\|_\infty^2}{\|\tilde{\mathcal{L}} - \mathcal{L}\|_F^2}\right). \tag{5.3}$$

SSIM measures the similarity of two images in luminance, contrast and structure, which is defined as

$$\text{SSIM}(\tilde{\mathbf{L}}, \mathbf{L}) := \frac{\left(2\mu_{\mathbf{L}}\mu_{\tilde{\mathbf{L}}} + c_1\right)\left(2\sigma_{\mathbf{L}\tilde{\mathbf{L}}} + c_2\right)}{\left(\mu_{\mathbf{L}}^2 + \mu_{\tilde{\mathbf{L}}}^2 + c_1\right)\left(\sigma_{\mathbf{L}}^2 + \sigma_{\tilde{\mathbf{L}}}^2 + c_2\right)}, \tag{5.4}$$

where $\mu_{\mathbf{L}}$ and $\mu_{\tilde{\mathbf{L}}}$ represent the mean values of the original data matrix $\mathbf{L}$ and the recovered data matrix $\tilde{\mathbf{L}}$, respectively, $\sigma_{\mathbf{L}}^2$ and $\sigma_{\tilde{\mathbf{L}}}^2$ represent the standard variances of $\mathbf{L}$ and $\tilde{\mathbf{L}}$, respectively, $\sigma_{\mathbf{L}\tilde{\mathbf{L}}}$ represents the covariance of $\mathbf{L}$ and $\tilde{\mathbf{L}}$, $c_1, c_2 > 0$ are two constants to stabilize the division with weak denominator. For tensor data, the SSIM value is obtained by calculating the average SSIM values for all frontal slices. As we can see, lower RSE value, higher PSNR and SSIM values mean better image quality.

5.3.1 Color Image

In this section, we select thirty color images from the Berkeley Segmentation Dataset (BSD) [33] for the test. These images are different natural scenes and objects in real life and all have size 321 × 481 × 3 or 481 × 321 × 3. For each image, we test three different noise levels: 30%, 20%, 10%. The parameter setting for color image recovery is listed in Table 1.

In Table 2, we list the average PSNR, SSIM, RSE values of the results recovered by different models for thirty color images. From the quantitative assessment results, we can see that the t-e^{γ}-$\mathcal{W}$ model always achieves the best recovery performance for different noise levels. For each image, the PSNR, SSIM and RSE obtained by different models are shown in Figs. 7 and 8, respectively. It can be seen that the results obtained by the proposed t-e^{γ}-$\mathcal{W}$ model reaches higher PSNR, SSIM and lower RSE values in most cases.

Table 2 The average PSNR, SSIM and RSE values for restoring results of different models for thirty color images corrupted by different noise levels

Noise level	Index	SNN	TNN	PSTNN	t-γ	t-e^{γ}	t-e^{γ}-$\mathcal{W}$
30%	PSNR	25.7517	26.7283	26.6856	28.5309	31.4776	**32.5711**
	SSIM	0.7620	0.8091	0.7877	0.8148	0.9218	**0.9249**
	RSE	0.1226	0.1089	0.1085	0.0896	0.0652	**0.0576**
20%	PSNR	27.4504	28.8543	30.0291	31.8479	36.4109	**37.3344**
	SSIM	0.8593	0.8988	0.9091	0.9307	0.9649	**0.9757**
	RSE	0.1008	0.0852	0.0744	0.0594	0.0368	**0.0330**
10%	PSNR	28.9491	30.9027	32.2806	34.5938	42.7264	**43.4984**
	SSIM	0.9089	0.9438	0.9529	0.9678	0.9901	**0.9925**
	RSE	0.0848	0.0674	0.0576	0.0434	0.0180	**0.0163**

The boldface number implies the best and the underline number indicates the second best

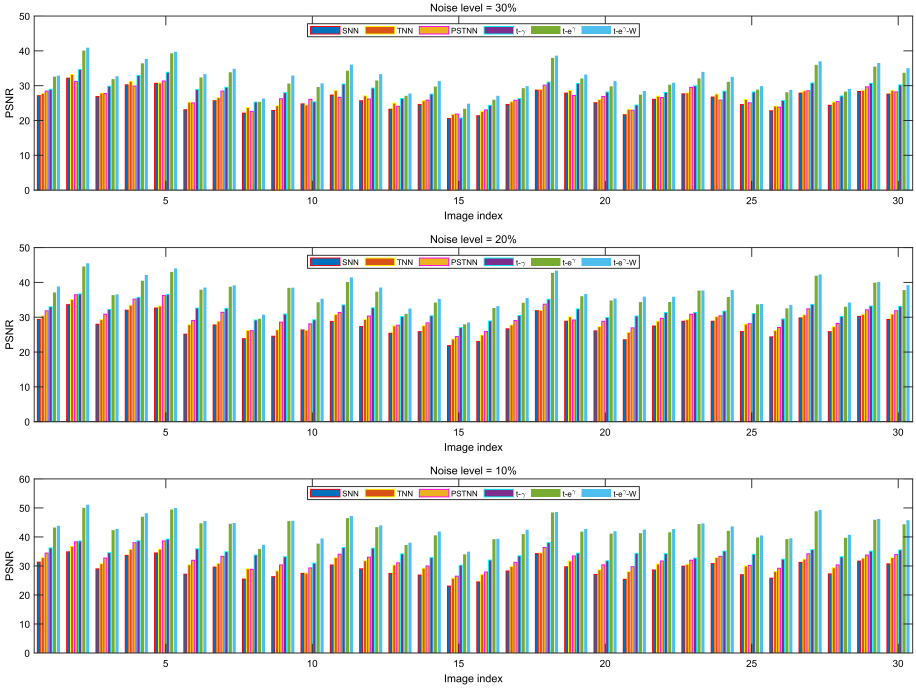


Fig. 7 The PSNR values obtained by different models for thirty color images with different noise levels. From top to bottom: the noise levels are 30%, 20% and 10%, respectively (Color figure online)

Figure 9 further exhibits the visual results on three sample color images, where only noise level 30% is shown exemplarily. From Fig. 9, we can see that the proposed t-e^{γ}-$\mathcal{W}$ model preserves image details (such as animal's eyes, the edge of the pyramid) better than other models.

Based on the above comparison, we have the following conclusions: First, among all competing models, the Tucker rank based SNN shows the worst recovery performance because the SNN is a loose convex relaxation of the sum of the Tucker rank. Second, the t-SVD based

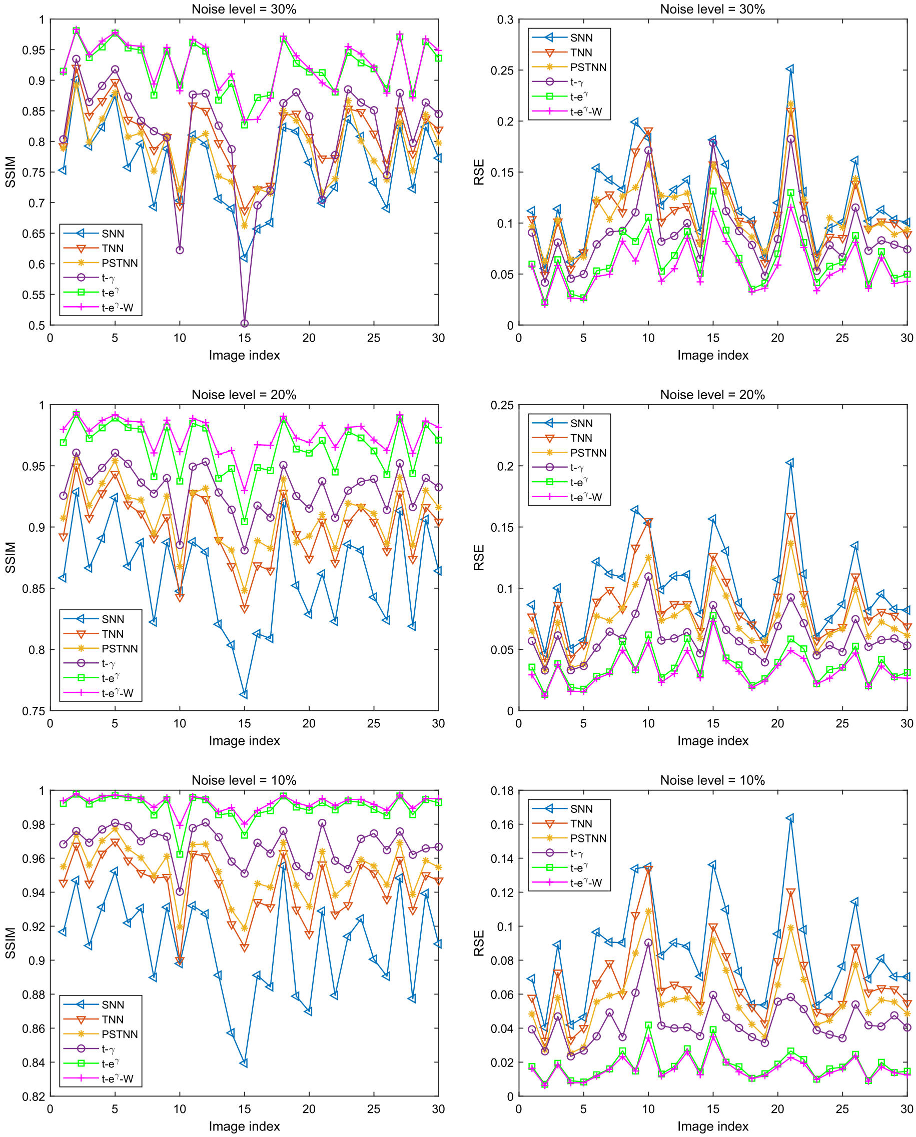
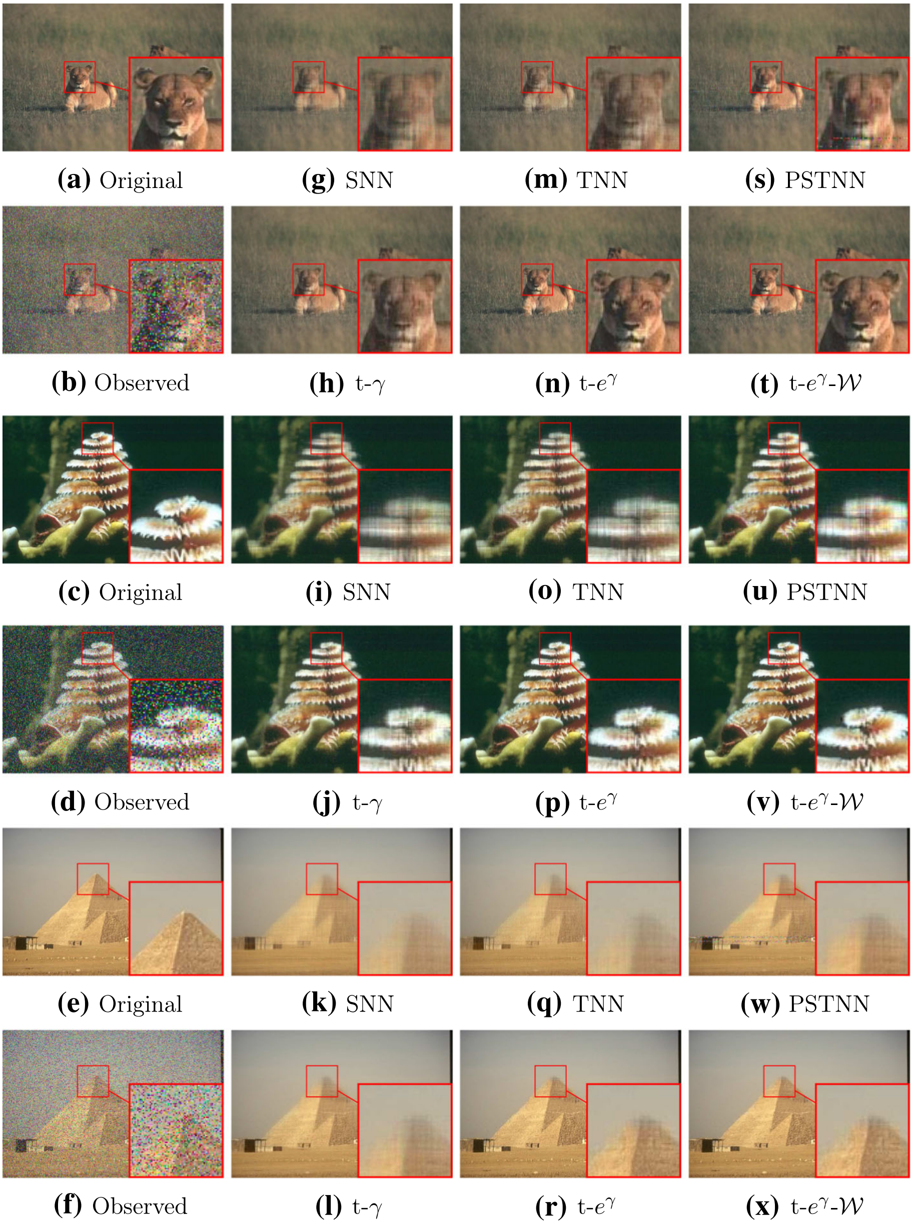


Fig. 8 The SSIM and RSE values obtained by different models for thirty color images with different noise levels. From top to bottom: the noise levels are 30%, 20% and 10%, respectively (Color figure online)

models can obtain better results because the t-SVD framework make more efficient use of the spatial structure of the tensor data. Third, among all competing non-convex models, t-e^γ and t-e^γ-$\mathcal{W}$ can achieve better recovery performance compared with PSTNN and t-γ. The reason is that both the tensor rank and the sparse constraint term in the t-e^γ and t-e^γ-$\mathcal{W}$ models are measured by the non-convex functions. This demonstrates the superiority of the non-convex models. Finally, the proposed t-e^γ-$\mathcal{W}$ can achieve the best performance and recover more details because the weighted ℓ_p-norm is a more accurate relaxation of the ℓ_0-norm compared with the ℓ_p-norm. This shows the effectiveness of the proposed model.



(a) Original **(g)** SNN **(m)** TNN **(s)** PSTNN

(b) Observed **(h)** t-γ **(n)** t-e^{γ} **(t)** t-e^{γ}-$\mathcal{W}$

(c) Original **(i)** SNN **(o)** TNN **(u)** PSTNN

(d) Observed **(j)** t-γ **(p)** t-e^{γ} **(v)** t-e^{γ}-$\mathcal{W}$

(e) Original **(k)** SNN **(q)** TNN **(w)** PSTNN

(f) Observed **(l)** t-γ **(r)** t-e^{γ} **(x)** t-e^{γ}-$\mathcal{W}$

Fig. 9 The recovered results on three sample color images with the noise level is 30%

Table 3 The parameter setting for brain MRI recovery

Noise level	Model	p	γ	λ
40%, 30%, 20%, 10%	SNN	–	–	[200, 2, 20]
	TNN	–	–	$\lambda_{\text{TNN}} \approx 0.005$
	PSTNN	–	–	$\lambda_{\text{TNN}} \approx 0.005$
	t-γ	–	$\frac{N_3}{\sqrt{\min\{N_1, N_2\}}} \approx 13.4536$	$\lambda_{\text{TNN}} \approx 0.005$
40%	t-e^{γ}	0.5	4	0.010
	t-e^{γ}-$\mathcal{W}$	0.9	4	0.005
30%	t-e^{γ}	0.5	3.9	0.010
	t-e^{γ}-$\mathcal{W}$	0.9	3.8	0.005
20%	t-e^{γ}	0.5	3.7	0.010
	t-e^{γ}-$\mathcal{W}$	0.9	3.7	0.005
10%	t-e^{γ}	0.6	3.1	0.010
	t-e^{γ}-$\mathcal{W}$	0.9	3.6	0.006

Table 4 The PSNR, SSIM and RSE values for restoring results of different models for MRI corrupted by different noise levels

Noise level	Index	SNN	TNN	PSTNN	t-γ	t-e^{γ}	t-e^{γ}-$\mathcal{W}$
40%	PSNR	20.7066	23.7179	16.3206	10.5769	$\underline{34.2151}$	**35.4236**
	SSIM	0.5818	0.4440	0.2678	0.0896	$\underline{0.9307}$	**0.9443**
	RSE	0.3089	0.2184	0.5134	0.9915	$\underline{0.0655}$	**0.0570**
30%	PSNR	21.7595	29.4889	24.8836	11.8221	$\underline{37.6826}$	**38.5430**
	SSIM	0.7236	0.7993	0.5312	0.1234	$\underline{0.9670}$	**0.9704**
	RSE	0.2736	0.1124	0.1916	0.8590	$\underline{0.0440}$	**0.0398**
20%	PSNR	22.6284	31.7847	34.5152	13.6897	$\underline{41.2392}$	**41.8638**
	SSIM	0.7931	0.9479	0.9105	0.1802	$\underline{0.9838}$	**0.9847**
	RSE	0.2476	0.0863	0.0631	0.6928	$\underline{0.0292}$	**0.0272**
10%	PSNR	23.4936	33.4494	37.8814	41.0253	$\underline{45.6415}$	**46.2240**
	SSIM	0.8398	0.9737	0.9817	0.9876	$\underline{0.9934}$	**0.9949**
	RSE	0.2241	0.0712	0.0428	0.0298	$\underline{0.0176}$	**0.0165**

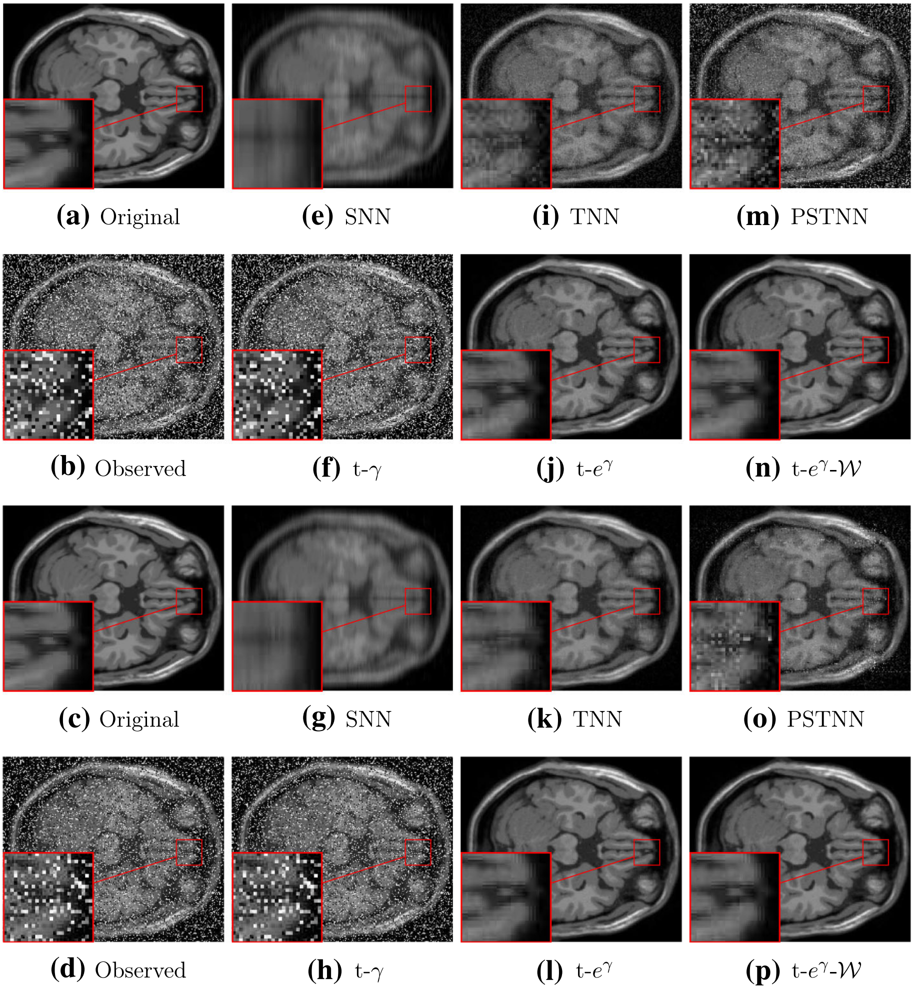
The boldface number implies the best and the underline number indicates the second best

5.3.2 Brain MRI

In this section, we conduct experiments on brain MRI[1], whose size is 181 × 217 × 181. The noise levels are set to be 40%, 30%, 20% and 10%, respectively. The parameter setting for brain MRI recovery is listed in Table 3.

Table 4 shows the numerical results obtained by six models for recovering the brain MRI. It can be seen that when the noise level is 40%, PSTNN and t-γ always yield unsatisfactory results. With the decrease of the noise levels, although the performance of SNN and TNN are increase steadily, they do not perform well for the lower noise levels (such as 10%). In contrast, t-e^{γ} and t-e^{γ}-$\mathcal{W}$ always have powerful recovery performance for different noise levels. In

[1] https://brainweb.bic.mni.mcgill.ca/.



(a) Original **(e)** SNN **(i)** TNN **(m)** PSTNN

(b) Observed **(f)** t-γ **(j)** t-e^{γ} **(n)** t-e^{γ}-$\mathcal{W}$

(c) Original **(g)** SNN **(k)** TNN **(o)** PSTNN

(d) Observed **(h)** t-γ **(l)** t-e^{γ} **(p)** t-e^{γ}-$\mathcal{W}$

Fig. 10 The recovered results on brain MRI with the noise levels are 40% and 30%, respectively. The first two rows: noise level is 40%. The last two rows: noise level is 30%

particular, the proposed t-e^{γ}-$\mathcal{W}$ model always achieves the best in terms of three assessment indexes for different noise levels, which further shows the robustness and unbiasedness of the proposed non-convex model.

Apart from quantitative assessment, we also show the visual comparison of one slice of the brain MRI recovered by different models in Fig. 10, where only noise levels 40% and 30% are shown exemplarily. Obviously, when the noise level is high, both the PSTNN and the t-γ fail to remove the noise. Although the SNN and TNN can remove most of the noise, the images obtained by them are very fuzzy. In contrast, the t-e^{γ} and t-e^{γ}-$\mathcal{W}$ can recover most information of the images while the images obtained by t-e^{γ}-$\mathcal{W}$ are almost the same as the original ones.

Table 5 The parameter setting for gray video recovery

Noise level	Model	p	γ	λ
40%, 30%, 10%	SNN	–	–	[200, 2, 20]
	TNN	–	–	$\lambda_{\text{TNN}} \approx 0.0062$
	PSTNN	–	–	$\lambda_{\text{TNN}} \approx 0.0062$
	t-γ	–	$\frac{N_3}{\sqrt{\min\{N_1, N_2\}}} \approx 12.50$	$\lambda_{\text{TNN}} \approx 0.0062$
40%	t-e^{γ}	0.8	1.3	0.005
	t-e^{γ}-$\mathcal{W}$	0.9	3	0.003
30%	t-e^{γ}	0.8	1.3	0.005
	t-e^{γ}-$\mathcal{W}$	0.9	2.8	0.003
10%	t-e^{γ}	0.7	1.1	0.004
	t-e^{γ}-$\mathcal{W}$	0.9	3	0.006

5.3.3 Gray Video Sequence

In this section, two gray videos[2] Grandma (144 × 176 × 150) and MotherDaughter (144 × 176 × 150) are chosen for the test. For each video, we test three different noise levels: 40%, 30%, 10%. The parameter setting for gray video recovery is listed in Table 5.

In Table 6, we report the PSNR, SSIM, and RSE values obtained by six models for the two gray videos in different noise levels. Figure 11 also exhibits the visual comparison on one frame for the two gray videos, where only noise level 40% is shown exemplarily. For different noise levels and the two videos, it can be observed from Table 6 that the t-e^{γ}-$\mathcal{W}$ always achieve best in terms of three assessment indexes. From Fig. 11, we can see that the recovery images by the SNN, TNN, PSTNN, and t-γ are very fuzzy while the t-e^{γ} and t-e^{γ}-$\mathcal{W}$ can obtain clear recovery images. In particular, owing to the two non-convex relaxations, the t-e^{γ}-$\mathcal{W}$ perfectly preserves more detailes such as the eyes and the mouth. The above comparison shows the feasibility and effectiveness of the proposed model.

5.4 Video Background Modeling

In this subsection, the proposed model is applied to solve the background modeling problem. The task of background modeling aims to separate the foreground objects from the background. This is the pre-process step in many vision applications such as surveillance video processing, moving target detection and segmentation [33]. Surveillance videos from a fixed camera can be naturally modeled by TRPCA since the frames of the background are highly correlated and thus can be considered as a low rank tensor and the moving foreground objects occupy only a small part of the image pixels and thus can be regarded as sparse errors. In our work, we select three color video sequences: highway (240 × 360), pedestrians (240 × 360) and PETS2006 (576 × 720) from CD.net dataset[3] in the baseline category, where the numbers in the parentheses denote the frame size. We only choose the frames from 900 to 1000 for these video sequences to reduce the computational time. For each sequence with color frame

[2] http://trace.eas.asu.edu/yuv/.

[3] http://jacarini.dinf.usherbrooke.ca/dataset2012/.

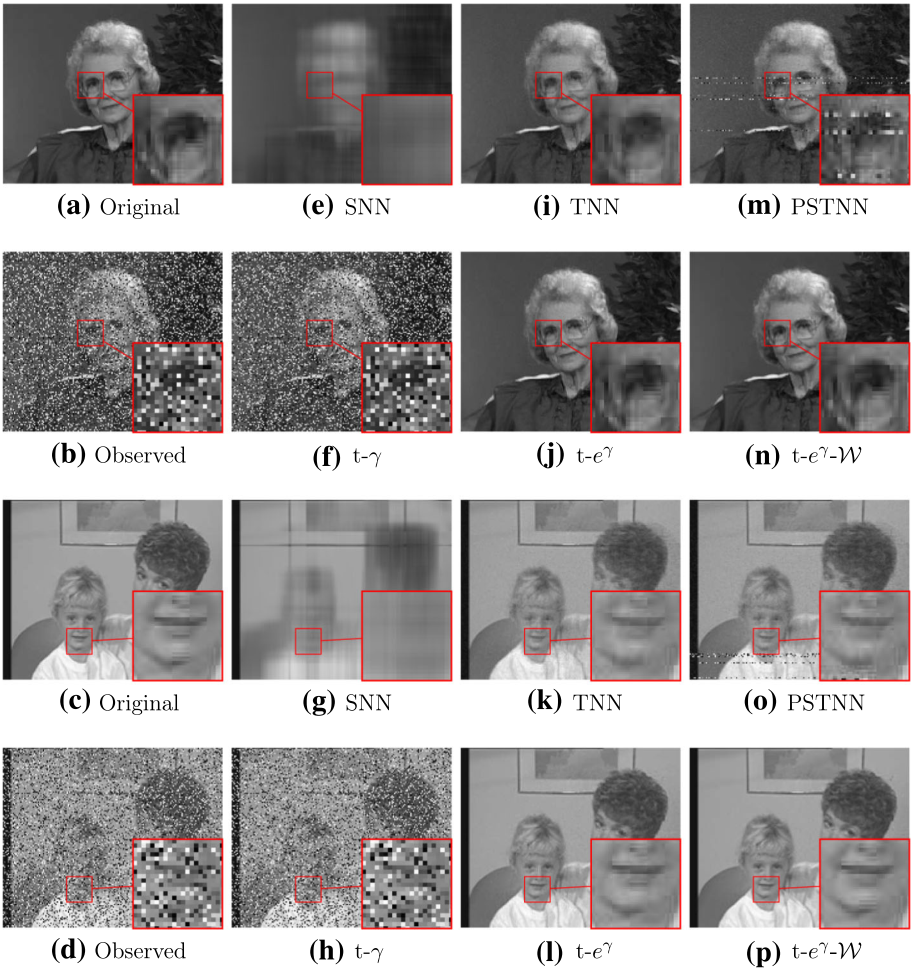
Table 6 The PSNR, SSIM and RSE values for restoring results of different models for gray videos corrupted by different noise levels

Data name	Noise level	Index	SNN	TNN	PSTNN	t-γ	t-e^{γ}	t-e^{γ}-$\mathcal{W}$
Grandma	40%	PSNR	21.0846	33.4939	26.6532	12.2612	37.7588	**40.0207**
		SSIM	0.6315	0.9354	0.8197	0.1440	0.9719	**0.9850**
		RSE	0.2730	0.0654	0.1438	0.7539	0.0399	**0.0308**
	30%	PSNR	21.8369	35.6890	33.1197	13.5227	38.3080	**42.4695**
		SSIM	0.6767	0.9684	0.9505	0.2258	0.9637	**0.9897**
		RSE	0.2503	0.0508	0.0683	0.6520	0.0376	**0.0233**
	10%	PSNR	23.2750	38.2877	39.6381	42.9750	46.9774	**48.2729**
		SSIM	0.7549	0.9830	0.9853	0.9913	0.9955	**0.9968**
		RSE	0.2121	0.0377	0.0322	0.0220	0.0139	**0.0119**
MotherDaughter	40%	PSNR	22.8801	31.6104	25.6515	13.3699	35.1288	**36.2370**
		SSIM	0.6738	0.9000	0.8338	0.1559	0.9497	**0.9662**
		RSE	0.1251	0.0458	0.0910	0.3740	0.0306	**0.0269**
	30%	PSNR	24.3051	33.5264	29.0472	14.6389	37.0461	**38.0702**
		SSIM	0.7115	0.9491	0.9203	0.2399	0.9654	**0.9751**
		RSE	0.1062	0.0367	0.0609	0.3231	0.0245	**0.0218**
	10%	PSNR	26.3732	35.4599	37.2766	39.0680	41.6703	**43.5535**
		SSIM	0.7724	0.9709	0.9744	0.9825	0.9905	**0.9917**
		RSE	0.0837	0.0294	0.0238	0.0194	0.0144	**0.0116**

Table 6 continued

Data name	Noise level	Index	SNN	TNN	PSTNN	t-γ	t-e^{γ}	t-e^{γ}-$\mathcal{W}$
Average	40%	PSNR	21.9824	32.5522	26.1524	12.8156	36.4438	**38.1288**
		SSIM	0.6527	0.9177	0.8267	0.1499	0.9608	**0.9756**
		RSE	0.1991	0.0556	0.1174	0.5639	0.0353	**0.0289**
	30%	PSNR	23.0710	34.6077	31.0835	14.0808	37.6771	**40.2698**
		SSIM	0.6941	0.9588	0.9354	0.2328	0.9646	**0.9824**
		RSE	0.1783	0.0438	0.0646	0.4876	0.0311	**0.0225**
	10%	PSNR	24.8241	36.8738	38.4574	41.0215	44.3239	**45.9132**
		SSIM	0.7636	0.9770	0.9798	0.9869	0.9930	**0.9943**
		RSE	0.1479	0.0335	0.0280	0.0207	0.0141	**0.0118**

The boldface number implies the best and the underline number indicates the second best


(a) Original (e) SNN (i) TNN (m) PSTNN

(b) Observed (f) t-γ (j) t-e^{γ} (n) t-e^{γ}-$\mathcal{W}$

(c) Original (g) SNN (k) TNN (o) PSTNN

(d) Observed (h) t-γ (l) t-e^{γ} (p) t-e^{γ}-$\mathcal{W}$

Fig. 11 The recovered results on gray videos with noise levels is 40%. The first two rows: Grandma. The last two rows: MotherDaughter

size $h \times w \times 3$ and frame number k, we reshape it to the $(hw) \times k \times 3$ tensor for the test. The parameter setting for video background modeling is listed in Table 7.

The F-Measure is used to assess the performance of the results of the background modeling, which is defined by

$$\text{F-Measure} = 2\frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}},$$

where

$$\text{Recall} = \frac{\text{the number of correctly classified foreground pixels}}{\text{the number of foreground pixels in ground truth}},$$
$$\text{Precision} = \frac{\text{the number of correctly classified foreground pixels}}{\text{the number of pixels classified as foreground}}.$$

Table 7 The parameter setting for Video Background Modeling

Data name	Model	p	γ	λ
highway pedestrians PETS2006	SNN	–	–	[200, 2, 20]
highway	TNN	–	–	$\lambda_{\text{TNN}} \approx 0.002$
	PSTNN	–	–	$\lambda_{\text{TNN}} \approx 0.002$
	t-γ	–	$\frac{N_3}{\sqrt{\min\{N_1, N_2\}}} \approx 0.2985$	$\lambda_{\text{TNN}} \approx 0.002$
	t-e^γ	0.1	1	0.01
	t-e^γ-$\mathcal{W}$	0.1	2	0.01
pedestrians	TNN	–	–	$\lambda_{\text{TNN}} \approx 0.002$
	PSTNN	–	–	$\lambda_{\text{TNN}} \approx 0.002$
	t-γ	–	$\frac{N_3}{\sqrt{\min\{N_1, N_2\}}} \approx 0.2985$	$\lambda_{\text{TNN}} \approx 0.002$
	t-e^γ	0.1	1	0.1
	t-e^γ-$\mathcal{W}$	0.1	2	0.1
PETS2006	TNN	–	–	$\lambda_{\text{TNN}} \approx$ 8.9652e-04
	PSTNN	–	–	$\lambda_{\text{TNN}} \approx$ 8.9652e-04
	t-γ	–	$\frac{N_3}{\sqrt{\min\{N_1, N_2\}}} \approx 0.2985$	$\lambda_{\text{TNN}} \approx$ 8.9652e-04
	t-e^γ	0.1	1	0.05
	t-e^γ-$\mathcal{W}$	0.1	2	0.05

Table 8 The F-Measure of the results of the background modeling obtained by different models

Video	SNN	TNN	PSTNN	t-γ	t-e^γ	t-e^γ-$\mathcal{W}$
highway	0.7879	0.8610	0.7959	0.8961	$\underline{0.8968}$	**0.8970**
pedestrians	0.9608	0.9607	0.8737	0.9634	$\underline{0.9794}$	**0.9825**
PETS2006	0.4077	0.7396	0.6807	0.8045	$\underline{0.8714}$	**0.8747**

In Table 8, we report the F-Measure obtained by different models for video background modeling. Obviously, compared with the SNN, the F-Measures of other competing models are high, which indicates that the t-SVD has a better performance in maintaining the spatial structure of the multi-dimensional data.

Figure 12 displays the background images and the foreground images separated by different models. We can see that the background images obtained by the SNN, TNN and PSTNN always have obvious foreground ghosting. It indicates that these models can not remove the sparse foreground component effectively. Although the t-γ model can get the clean background images in the highway and pedestrians videos, a twisted foreground image can be found in the PETS2006 video. In addition, we can see that the results obtained by the t-e^γ and t-e^γ-$\mathcal{W}$ models are almost same. But from Table 8, the background modeling results obtained by the t-e^γ-$\mathcal{W}$ model outperform those obtained by the t-e^γ model in terms of F-Measure.

In summary, the proposed model shows strong background modeling capability, producing satisfying results both in visualization and in quantitative assessment.

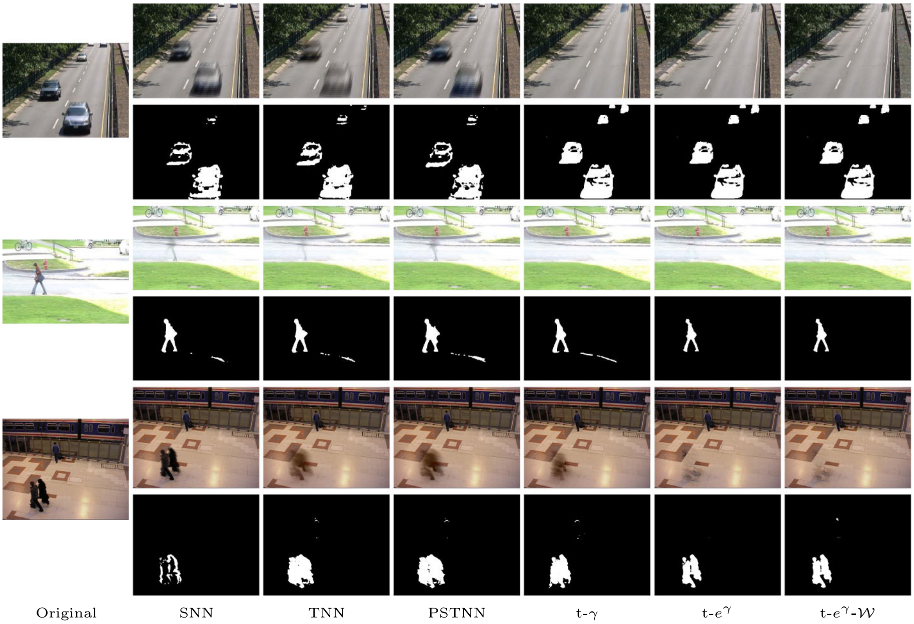


Fig. 12 The background modeling results obtained by different models. From left to right: original frames, the background modeling results obtained by the SNN, TNN, PSTNN, t-γ, t-e^{γ}, and t-e^{γ}-$\mathcal{W}$, respectively. Rows 1 and 2 are samples of highway; rows 3 and 4 are samples of pedestrians; rows 5 and 6 are samples of PETS2006

6 Concluding Remarks

In this paper, we present a new non-convex approach for the TRPCA problem. Different from existing non-convex TRPCA models, we not only propose a novel non-convex relaxation for the tensor tubal rank but also introduce a new and suitable non-convex sparsity measuration for the sparse constraint term rather than by the ℓ_1-norm commonly used in literature. It is well known that convergence analysis of the non-convex optimization algorithm is a challenging problem, and we are able to show that the sequence generated by the proposed algorithm always converges to a KKT stationary point. The numerical experiments for practical applications are given to demonstrate that the proposed non-convex model outperforms the existing approaches.

Acknowledgements The authors would like to thank three anonymous referees for their very helpful comments and suggestions.

Data Availability The authors confirm that the data supporting the findings of this study are available within the article and its supplementary materials.

References

1. Bengua, J.A., Phien, H.N., Tuan, H.D., Do, M.N.: Efficient tensor completion for color image and video recovery: low-rank tensor train. IEEE Trans. Image Process. **26**(5), 2466–2479 (2017)
2. Cai, S.T., Luo, Q.L., Yang, M., Li, W., Xiao, M.Q.: Tensor robust principal component analysis via nonconvex low rank approximation. Appl. Sci. **9**(7), 1411 (2019)
3. Candès, E.J., Wakin, M.B., Boyd, S.P.: Enhancing sparsity by reweighted ℓ_1 minimization. J. Fourier Anal. Appl. **14**(5), 877–905 (2007)
4. Candès, E.J., Li, X., Ma, Y., Wright, J.: Robust principal component analysis? J. ACM **58**(3), 1–37 (2011)
5. Chartrand, R.: Exact reconstructions of sparse signals via nonconvex minimization. IEEE Signal Process. Lett. **14**(10), 707–710 (2007)
6. Chen, J.K.: A new model of tensor robust principal component analysis and its application [D], pp. 1–49. South China Normal University, Guangzhou (2020)
7. Deisenroth, M.P., Faisal, A.A., Ong, C.S.: Mathematics for Machine Learning. Cambridge University Press, Cambridge (2019)
8. De Lathauwer, L., De Moor, B., Vandewalle, J.: A multilinear singular value decomposition. SIAM J. Matrix Anal. Appl. **21**(4), 1253–1278 (2000)
9. Donoho, D.L.: De-noising by soft-thresholding. IEEE Trans. Inform. Theory **41**(3), 613–627 (1995)
10. Dong, W.S., Shi, G.M., Li, X., Ma, Y., Huang, F.: Compressive sensing via nonlocal low-rank regularization. IEEE Trans. Image Process. **23**(8), 3618–3632 (2014)
11. Feng, L.L., Liu, Y.P., Chen, L.X., Zhang, X., Zhu, C.: Robust block tensor principal component analysis. Signal Process. **166**, 107271 (2020)
12. Gao, S.Q., Zhuang, X.H.: Robust approximations of low-rank minimization for tensor completion. Neurocomputing **379**, 319–333 (2020)
13. Georghiades, A.S., Belhumeur, P.N., Kriegman, D.J.: 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 (2002)
14. Gu, S.H., Xie, Q., Meng, D.Y., Zuo, W.M., Feng, X.C., Zhang, L.: Weighted nuclear norm minimization and its applications to low level vision. Int. J. Comput. Vis. **121**(2), 183–208 (2017)
15. Hitchcock, F.L.: The expression of a tensor or a polyadic as a sum of products. Stud. Appl. Math. **6**(1–4), 164–189 (1927)
16. Jiang, T.X., Huang, T.Z., Deng, L.J.: Multi-dimensional imaging data recovery via minimizing the partial sum of tubal nuclear norm. J. Comput. Appl. Math. **372**, 112680 (2020)
17. Ji, T.Y., Huang, T.Z., Zhao, X.L., Ma, T.H., Deng, L.J.: A non-convex tensor rank approximation for tensor completion. Appl. Math. Model. **48**, 410–422 (2017)
18. Kang, Z., Peng, C., Cheng, Q.: Robust PCA via nonconvex rank approximation. In: IEEE International Conference on Data Mining, pp. 211–220 (2015)
19. Karlsson, L., Kressner, D., Uschmajew, A.: Parallel algorithms for tensor completion in the CP format. Parallel Comput. **57**, 222–234 (2016)
20. Kilmer, M.E., Martin, C.D.: Factorization strategies for third-order tensors. Linear Algebra Appl. **435**(3), 641–658 (2011)
21. Kilmer, M.E., Braman, K., Hao, N., Hoover, R.C.: 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 (2013)
22. Kolda, T.G., Bader, B.W.: Tensor decompositions and applications. SIAM Rev. **51**(3), 455–500 (2009)
23. Lewis, A.S., Sendov, H.S.: Nonsmooth analysis of singular values. Part I: theory. Set-Valued Anal. **13**(3), 213–241 (2005)
24. Liu, G.C., Lin, Z.C., Yan, S.C., Sun, J., Yu, Y., Ma, Y.: Robust recovery of subspace structures by low-rank representation. IEEE Trans. Pattern Anal. Mach. Intell. **35**(1), 171–184 (2010)
25. Liu, J., Musialski, P., Wonka, P., Ye, J.P.: Tensor completion for estimating missing values in visual data. IEEE Trans. Pattern Anal. Mach. Intell. **35**(1), 208–220 (2013)
26. Liu, Y.Y., Zhao, X.L., Zheng, Y.B., Ma, T.H., Zhang, H.Y.: Hyperspectral image restoration by tensor fibered rank constrained optimization and plug-and-play regularization. IEEE Trans. Geosci. Remote Sens. https://doi.org/10.1109/TGRS.2020.3045169 (2021)
27. Lou, J., Cheung, Y.M.: Robust low-rank tensor minimization via a new tensor spectral k-support norm. IEEE Trans. Image Process. **29**, 2314–2327 (2020)
28. Li, X.T., Zhao, X.L., Jiang, T.X., Zheng, Y.B., Ji, T.Y., Huang, T.Z.: Low-rank tensor completion via combined non-local self-similarity and low-rank regularization. Neurocomputing **367**(20), 1–12 (2019)
29. Lu, C.Y., Feng, J.S., Chen, Y.D., Liu, W., Lin, Z.C., Yan, S.C.: Tensor robust principal component analysis with a new tensor nuclear norm. IEEE Trans. Pattern Anal. Mach. Intell. **42**(4), 925–938 (2020)

30. Luenberger, D.G., Ye, Y.Y.: Linear and Nonlinear Programming. Springer, Switzerland (2015)
31. Lu, H.P., Plataniotis, K.N., Venetsanopoulos, A.N.: MPCA: multilinear principal component analysis of tensor objects. IEEE Trans. Neural Network **19**(1), 18–39 (2008)
32. Lu, Z.S.: Iterative reweighted minimization methods for l_p regularized unconstrained nonlinear programming. Math. Program. **147**(1–2), 277–307 (2014)
33. Martin, D., Fowlkes, C., Tal, D., Malik, J.: 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, vol. 2, pp. 416–423 (2001)
34. Mørup, M.: Applications of tensor (multiway array) factorizations and decompositions in data mining. Wiley Interdiscip. Rev. Data Min. Knowl. Discov. **1**(1), 24–40 (2011)
35. Mu, Y., Wang, P., Lu, L.F., Zhang, X.Y., Qi, L.Y.: Weighted tensor nuclear norm minimization for tensor completion using tensor-SVD. Pattern Recogn. Lett. **130**, 4–11 (2020)
36. Oseledets, I.V.: Tensor-train decomposition. SIAM J. Sci. Comput. **33**(5), 2295–2317 (2011)
37. Pan, P., Wang, Y.L., Chen, Y.Y., Wang, S.Q., He, G.P.: A new nonconvex rank approximation of RPCA. Sci. Tech. Eng. **17**(31), 1671–1815 (2017)
38. Qi, L.Q., Luo, Z.Y.: Tensor Analysis: Spectral Theory and Special Tensors. SIAM Press, Philadephia (2017)
39. Qi, N., Shi, Y.H., Sun, X.Y., Wang, J.D., Yin, B.C., Gao, J.B.: Multi-dimensional sparse models. IEEE Trans. Pattern Recogn. Mach. Intell. **40**(1), 163–178 (2018)
40. Silva, V.D., Lim, L.H.: Tensor rank and the ill-posedness of the best low-rank approximation problem. SIAM J. Matrix Anal. Appl. **30**(3), 1084–1127 (2008)
41. Song, G.J., Ng, M.K., Zhang, X.J.: Robust tensor completion using transformed tensor singular value decomposition. Numer. Linear Algeb. Appl. **27**(3), e2299 (2020)
42. Xue, J.Z., Zhao, Y.Q., Liao, W.Z., Chan, C.W.: Nonconvex tensor rank minimization and its applications to tensor recovery. Inf. Sci. **503**, 109–128 (2019)
43. Xu, W.H., Zhao, X.L., Ji, T.Y., Miao, J.Q., Ma, T.H., Wang, S., Huang, T.Z.: Laplace function based nonconvex surrogate for low-rank tensor completion. Signal Process. Image Commun. **73**, 62–69 (2019)
44. Xu, Z.B., Zhang, H., Wang, Y., Chang, X.Y., Liang, Y.: $L_{1/2}$ regularization. Sci. China Inf. Sci. **53**(6), 1159–1169 (2010)
45. Yang, J.H., Zhao, X.L., Ji, T.Y., Ma, T.H., Huang, T.Z.: Low-rank tensor train for tensor robust principal component analysis. Appl. Math. Comput. **367**(15), 124783 (2020)
46. Yang, M., Luo, Q.L., Li, W., Xiao, M.Q.: Multiview clustering of images with tensor rank minimization via nonconvex approach. SIAM J. Imaging Sci. **13**(4), 2361–2392 (2020)
47. Zhang, Z.M., Aeron, S.: Exact tensor completion using t-SVD. IEEE Trans. Signal Process. **65**(6), 1511–1526 (2017)
48. Zhao, X.L., Xu, W.H., Jiang, T.X., Wang, Y., Ng, M.K.: Deep plug-and-play prior for low-rank tensor completion. Neurocomputing **400**, 137–149 (2020)
49. Zhao, X.Y., Bai, M.R., Ng, M.K.: Nonconvex optimization for robust tensor completion from grossly sparse observations. J. Sci. Comput. **85**(2), 1–32 (2020)
50. Zhao, Y.B.: Reweighted ℓ_1-minimization for sparse solutions to underdetermined linear systems. SIAM J. Optim. **22**(3), 1065–1088 (2012)
51. Zheng, Y.B., Huang, T.Z., Zhao, X.L., Jiang, T.X., Ma, T.H., Ji, T.Y.: Mixed noise removal in hyperspectral image via low-fibered-rank regularization. IEEE Trans. Geosci. Remote Sens. **58**(1), 734–749 (2020)
52. Zheng, Y.B., Huang, T.Z., Zhao, X.L., Zhao, Q.B., Jiang, T.X.: Fully-connected tensor network decomposition and its application to higher-order tensor completion. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35(12), pp. 11071–11078 (2021)
53. Zhou, M.Y., Liu, Y.P., Long, Z., Chen, L.X., Zhu, C.: Tensor rank learning in CP decomposition via convolutional neural network. Signal Process Image Commun. **73**, 12–21 (2019)
54. Zuo, W.M., Meng, D.Y., Zhang, L., Feng, X.C., Zhang, D.: A generalized iterated shrinkage algorithm for nonconvex sparse coding. In: IEEE International Conference on Computer Vision, pp. 217–224 (2013)