# Low-rank Tensor Completion with a New Tensor Nuclear Norm Induced by Invertible Linear Transforms

Canyi Lu[1], Xi Peng[2], Yunchao Wei[3,4]
[1] Carnegie Mellon University, [2] Sichuan University, [3] University of Illinois at Urbana-Champaign, [4] University of Technology Sydney
canyilu@gmail.com, pengx.gm@gmail.com, wychao1987@gmail.com

## Abstract

*This work studies the low-rank tensor completion problem, which aims to exactly recover a low-rank tensor from partially observed entries. Our model is inspired by the recently proposed tensor-tensor product (t-product) [9] based on any invertible linear transforms. When the linear transforms satisfy certain conditions, we deduce the new tensor tubal rank, tensor spectral norm, and tensor nuclear norm. Equipped with the tensor nuclear norm, we then solve the tensor completion problem by solving a convex program and provide the theoretical bound for the exact recovery under certain tensor incoherence conditions. The achieved sampling complexity is order-wise optimal. Our model and result greatly extend existing results in the low-rank matrix [5] and tensor completion [16]. Numerical experiments verify our results and the application on image recovery demonstrates the superiority of our method.*

## 1. Introduction

With the availability of cheap memory and the advances in modern computer technology, it is now possible to collect, store and process more data for science and engineering applications than ever before. The real data are usually multidimensional in nature: the information is stored in multiway arrays, known as tensors [26], whose entries are indexed by several continuous or discrete variables. For example, a color image is a 3-way object with column, row and color modes; a greyscale video is indexed by two spatial variables and one temporal variable. There are many applications which involve the tensor representation and operation, including signal processing [6], computer vision [27], data mining [23], and many others. A common approach in these applications is to manipulate the tensor data by taking the advantage of its multi-dimensional structure. Collapsing the multiway data to matrices usually leads to information loss and would cause performance degradation. It is observed that the real tensor data are often of extremely high-dimension. But the tensor of interest is frequently low-rank, or approximately so [11], and hence has a much lower-dimensional structure. This motivates the low-rank tensor estimation and recovery problem which is gaining significant attention in many different areas, both theoretically and practically: e.g., estimating latent variable graphical models [1], classifying audio [20], image and video completion [13, 16], to name a few.

In this paper, we focus on the low-rank tensor completion problem, which aims to exactly recover a low-rank tensor from an incomplete observation. Such a problem can be regarded as an extension of the low-rank matrix completion problem [3, 5], which has been applied to image denoising [18] and multi-label image classification [2]. It is shown that under certain incoherence conditions, the rank $r$ matrix $\boldsymbol{M} \in \mathbb{R}^{n\times n}$ with $O(nr\log^2 n)$ observations, can be recovered with high probability by solving the following convex model [5]

$$\min_{\boldsymbol{X}} \|\boldsymbol{X}\|_*, \text{ s.t. } \boldsymbol{P}_{\boldsymbol{\Omega}}(\boldsymbol{X}) = \boldsymbol{P}_{\boldsymbol{\Omega}}(\boldsymbol{M}), \tag{1}$$

where $\boldsymbol{P}_{\boldsymbol{\Omega}}(\boldsymbol{X})$ denotes the projection of $\boldsymbol{X}$ on the observed set $\boldsymbol{\Omega}$, and $\|\boldsymbol{X}\|_*$ is the nuclear norm of $\boldsymbol{X}$, defined as the sum of its singular values. The nuclear norm is the convex envelope of the matrix rank within a certain set. This leads to the order-wise optimal sampling complexity $O(nr\log^2 n)$ for (1), compared with the degrees of freedom $O(nr)$ for a rank $r$ matrix [5].

It is expected to extend the low-rank matrix completion model and analysis to the tensor case. There have been several tensor recovery models proposed based on different tensor rank definitions and convex surrogates. But they have some limitations in real applications. The CP rank [11], defined as the smallest number of rank one tensor decomposition, is NP-hard to compute. Its convex envelope is not clear. This makes the low CP rank tensor recovery challenging. To avoid this issue, the tractable Tucker rank [11] and its convex relaxation are more widely used. The Sum of Nuclear Norms (SNN) [13] is used as a convex surrogate

978-1-7281-3293-8/19/$31.00 ©2019 IEEE
DOI 10.1109/CVPR.2019.00615

and applied for Low-Rank Tensor Completion (LRTC) [21]

$$\min_{\boldsymbol{\mathcal{X}}} \sum_{i=1}^{k} \lambda_i \|\boldsymbol{X}^{\{i\}}\|_*, \text{ s.t. } \mathcal{P}_{\boldsymbol{\Omega}}(\boldsymbol{\mathcal{X}}) = \mathcal{P}_{\boldsymbol{\Omega}}(\boldsymbol{\mathcal{M}}), \quad (2)$$

where $\boldsymbol{X}^{\{i\}}$ is the mode-$i$ matricization of $\boldsymbol{\mathcal{X}}$ [11], $\mathcal{P}_{\boldsymbol{\Omega}}(\boldsymbol{\mathcal{X}})$ denotes the projection of $\boldsymbol{\mathcal{X}}$ on the observed set $\boldsymbol{\Omega}$, and $\lambda_i > 0$. The effectiveness of this approach for image processing has been well studied in [13, 7, 25, 24]. However, SNN is not the tightest convex relaxation of the Tucker rank [22]. In theory, the above model can be substantially suboptimal [21] as the required number of measurements is much higher than the degrees of freedom of tensor with Tucker rank $(r, r, \cdots, r)$. This is different from the low-rank matrix completion model using the nuclear norm minimization which owns the optimal recovery rate [5].

More recently, based on the tensor-tensor product (t-product) and tensor SVD (t-SVD) [10], a new tensor nuclear norm is proposed and applied in tensor completion [16] and tensor robust PCA [14, 15]. The main advantage of the t-product induced tensor nuclear norm based models is that they own the same tight recovery bound as the matrix cases [16]. In [9], the authors observe that the t-product is based on a convolution-like operation, which can be implemented using the Discrete Fourier Transform (DFT). In order to properly motivate this transform based approach, a more general tensor-tensor product definition is proposed based on any invertible linear transforms. The transform based t-product also owns a matrix-algebra based interpretation in the spirit of [8]. Such a new transform based t-product is of great interest in practice as it allows to use different linear transforms for different real data formatted as tensors. In this work, we focus on the important low-rank tensor recovery problem based on the transform induced t-product. Specifically, we are interested in the following questions:

- How to define a new tensor rank and tensor nuclear norm induced by the linear transform based t-product?
- Is there any restriction on the choice of the invertible linear transform?
- With a new linear transform based tensor nuclear norm, is there any corresponding recovery guarantee for low-rank tensor completion?

The main contributions of this work are to solve the above problems. We show that if the linear transform satisfies certain condition, then a new tensor nuclear norm can be defined induced by the transform based t-product. We can further define the same tensor tubal rank as in [10] based on the t-SVD. Finally, we study the low tubal rank tensor recovery problem and solve it by the transform based tensor nuclear norm minimization which is convex. In theory, we prove that under certain tensor incoherence conditions, the underlying low tubal rank tensor can be exactly recovered with high probability by convex optimization. Our model and result are much more general than [16], since we have much more general choices of the invertible linear transforms. Our proof of the main result is much more challenging since the transform based t-product has no equivalent formulation in the original domain. This is quite different from the t-product [10] which is defined based on the block circulant matrix structure in the original domain. Such a structure is important in the proofs in existing works, e.g., [16].

The rest of this paper is structured as follows. Section 2 gives some notations and presents the new tensor nuclear norm induced by the transform based t-product. Section 3 solves the low-rank tensor completion problem by minimizing the proposed tensor nuclear norm and provides the exact recovery guarantee in theory. Numerical experiments conducted on both synthesis and real world data are presented in Section 4. We finally conclude this work in Section 5.

## 2. A New Tensor Nuclear Norm Induced by Transforms based T-product

In this section, we introduce some notations and definitions used in this paper. Some of these are from [10, 15].

### 2.1. Notations

We denote scalars by lowercase letters, e.g., $a$, vector by boldface lowercase letters, e.g., $\boldsymbol{a}$, matrices by boldface capital letters, e.g., $\boldsymbol{A}$, and tensors by boldface Euler script letters, e.g., $\boldsymbol{\mathcal{A}}$. For a 3-way tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, we denote its $(i, j, k)$-th entry as $\boldsymbol{\mathcal{A}}_{ijk}$ or $a_{ijk}$ and use the Matlab notation $\boldsymbol{\mathcal{A}}(i, :, :)$, $\boldsymbol{\mathcal{A}}(:, i, :)$ and $\boldsymbol{\mathcal{A}}(:, :, i)$ to denote respectively the $i$-th horizontal, lateral and frontal slice [11]. More often, the frontal slice $\boldsymbol{\mathcal{A}}(:, :, i)$ is denoted compactly as $\boldsymbol{A}^{(i)}$. The tube is denoted as $\boldsymbol{\mathcal{A}}(i, j, :)$. The inner product between $\boldsymbol{A}$ and $\boldsymbol{B}$ in $\mathbb{R}^{n_1 \times n_2}$ is defined as $\langle \boldsymbol{A}, \boldsymbol{B} \rangle = \mathtt{Tr}(\boldsymbol{A}^\top \boldsymbol{B})$, where $\boldsymbol{A}^\top$ denotes the transpose of $\boldsymbol{A}$ and $\mathtt{Tr}(\cdot)$ denotes the matrix trace. The inner product between $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{B}}$ in $\mathbb{R}^{n_1 \times n_2 \times n_3}$ is defined as $\langle \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}} \rangle = \sum_{i=1}^{n_3} \left\langle \boldsymbol{A}^{(i)}, \boldsymbol{B}^{(i)} \right\rangle$. We denote $\boldsymbol{I}_n$ as the $n \times n$ sized identity matrix.

Some norms of vector, matrix and tensor are used. We denote the $\ell_1$-norm as $\|\boldsymbol{\mathcal{A}}\|_1 = \sum_{ijk} |a_{ijk}|$, the infinity norm as $\|\boldsymbol{\mathcal{A}}\|_\infty = \max_{ijk} |a_{ijk}|$ and the Frobenius norm as $\|\boldsymbol{\mathcal{A}}\|_F = \sqrt{\sum_{ijk} a_{ijk}^2}$, respectively. The above norms reduce to the vector or matrix norms if $\boldsymbol{\mathcal{A}}$ is a vector or a matrix. For $\boldsymbol{v} \in \mathbb{R}^n$, the $\ell_2$-norm is $\|\boldsymbol{v}\|_2 = \sqrt{\sum_i v_i^2}$. The spectral norm of a matrix $\boldsymbol{A}$ is denoted as $\|\boldsymbol{A}\| = \max_i \sigma_i(\boldsymbol{A})$, where $\sigma_i(\boldsymbol{A})$'s are the singular values of $\boldsymbol{A}$. The matrix nuclear norm is $\|\boldsymbol{A}\|_* = \sum_i \sigma_i(\boldsymbol{A})$.

### 2.2. Transform based T-product Induced Tensor Nuclear Norm

The work [10] gives the first definition of tensor-tensor product. For $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, we define the block circulant matrix $\texttt{bcirc}(\mathcal{A}) \in \mathbb{R}^{n_1 n_3 \times n_2 n_3}$ of $\mathcal{A}$ as

$$\texttt{bcirc}(\mathcal{A}) = \begin{bmatrix} \boldsymbol{A}^{(1)} & \boldsymbol{A}^{(n_3)} & \cdots & \boldsymbol{A}^{(2)} \\ \boldsymbol{A}^{(2)} & \boldsymbol{A}^{(1)} & \cdots & \boldsymbol{A}^{(3)} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{A}^{(n_3)} & \boldsymbol{A}^{(n_3-1)} & \cdots & \boldsymbol{A}^{(1)} \end{bmatrix}.$$

We also need the following two operators

$$\texttt{unfold}(\mathcal{A}) = \begin{bmatrix} \boldsymbol{A}^{(1)} \\ \boldsymbol{A}^{(2)} \\ \vdots \\ \boldsymbol{A}^{(n_3)} \end{bmatrix}, \ \texttt{fold}(\texttt{unfold}(\mathcal{A})) = \mathcal{A},$$

where the $\texttt{unfold}$ operator maps $\mathcal{A}$ to a matrix of size $n_1 n_3 \times n_2$ and $\texttt{fold}$ is its inverse operator. Let $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{B} \in \mathbb{R}^{n_2 \times l \times n_3}$. Then the t-product $\mathcal{C} = \mathcal{A} * \mathcal{B}$ is defined to be a tensor of size $n_1 \times l \times n_3$,

$$\mathcal{C} = \mathcal{A} * \mathcal{B} = \texttt{fold}(\texttt{bcirc}(\mathcal{A}) \cdot \texttt{unfold}(\mathcal{B})). \quad (3)$$

The block circulant matrix can be block diagonalized using Discrete Fourier Transform (DFT) matrix $\boldsymbol{F}_{n_3}$, i.e.,

$$(\boldsymbol{F}_{n_3} \otimes \boldsymbol{I}_{n_1}) \cdot \texttt{bcirc}(\mathcal{A}) \cdot (\boldsymbol{F}_{n_3}^{-1} \otimes \boldsymbol{I}_{n_2}) = \bar{\boldsymbol{A}}, \quad (4)$$

where $\otimes$ denotes the Kronecker product, and $\bar{\boldsymbol{A}}$ is a block diagonal matrix with the $i$-th block $\bar{\boldsymbol{A}}^{(i)}$ being the $i$-th frontal slices of $\bar{\mathcal{A}}$ obtained by performing DFT of $\mathcal{A}$ along the 3-rd dimension, i.e., $\bar{\mathcal{A}} = \texttt{fft}(\mathcal{A}, [\,], 3)$ by using the Matlab command $\texttt{fft}$. We denote

$$\mathcal{D} = \mathcal{A} \odot \mathcal{B} \quad (5)$$

as the frontal-slice-wise product (Definition 2.1 in [9]), i.e.,

$$\boldsymbol{D}^{(i)} = \boldsymbol{A}^{(i)} \boldsymbol{B}^{(i)}, \ i = 1, \cdots, n_3. \quad (6)$$

Then the block diagonalized property in (4) implies that $\bar{\mathcal{C}} = \bar{\mathcal{A}} \odot \bar{\mathcal{B}}$. So the t-product in (3) is equivalent to the matrix-matrix product in the transform domain using DFT.

In [9], the authors propose a more general definition of t-product based on any invertible linear transform $L$. In this work, we consider the linear transform $L : \mathbb{R}^{n_1 \times n_2 \times n_3} \to \mathbb{R}^{n_1 \times n_2 \times n_3}$ which gives $\bar{\mathcal{A}}$ in terms of transforms applied to each tube fiber $\mathcal{A}(i, j, :)$. Or we have

$$\bar{\mathcal{A}} = L(\mathcal{A}) = \mathcal{A} \times_3 \boldsymbol{L}, \quad (7)$$

**Algorithm 1** Transform $L$ based t-product [9]

**Input:** $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, $\mathcal{B} \in \mathbb{R}^{n_2 \times l \times n_3}$, and $L : \mathbb{R}^{n_1 \times n_2 \times n_3} \to \mathbb{R}^{n_1 \times n_2 \times n_3}$.
**Output:** $\mathcal{C} = \mathcal{A} *_L \mathcal{B} \in \mathbb{R}^{n_1 \times l \times n_3}$.

1. Compute $\bar{\mathcal{A}} = L(\mathcal{A})$ and $\bar{\mathcal{B}} = L(\mathcal{B})$.
2. Compute each frontal slice of $\bar{\mathcal{C}}$ by
$$\bar{\boldsymbol{C}}^{(i)} = \bar{\boldsymbol{A}}^{(i)} \bar{\boldsymbol{B}}^{(i)}, \ i = 1, \cdots, n_3$$
3. Compute $\mathcal{C} = L^{-1}(\bar{\mathcal{C}})$.

where $\times_3$ denotes the mode-3 product (Definition 2.5 in [9]), and $\boldsymbol{L} \in \mathbb{R}^{n_3 \times n_3}$ can be arbitrary invertible matrix[1]. Similarly, we have the inverse mapping

$$L^{-1}(\mathcal{A}) = \mathcal{A} \times_3 \boldsymbol{L}^{-1}. \quad (8)$$

Then the transform $L$ based t-product is defined as follows.

**Definition 2.1.** ***(T-product)*** *[9] Let $L$ be any invertible linear transform in (7), and $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{B} \in \mathbb{R}^{n_2 \times l \times n_3}$. Then the transform $L$ based t-product, denoted as $\mathcal{C} = \mathcal{A} *_L \mathcal{B}$, is defined such that $L(\mathcal{C}) = L(\mathcal{A}) \odot L(\mathcal{B})$.*

We denote $\bar{\boldsymbol{A}} \in \mathbb{R}^{n_1 n_3 \times n_2 n_3}$ as a block diagonal matrix with its $i$-th block on the diagonal as the $i$-th frontal slice $\bar{\boldsymbol{A}}^{(i)}$ of $\bar{\mathcal{A}} = L(\mathcal{A})$, i.e.,

$$\bar{\boldsymbol{A}} = \texttt{bdiag}(\bar{\mathcal{A}}) = \begin{bmatrix} \bar{\boldsymbol{A}}^{(1)} & & & \\ & \bar{\boldsymbol{A}}^{(2)} & & \\ & & \ddots & \\ & & & \bar{\boldsymbol{A}}^{(n_3)} \end{bmatrix},$$

where $\texttt{bdiag}$ is an operator which maps tensor $\bar{\mathcal{A}}$ to the block diagonal matrix $\bar{\boldsymbol{A}}$. Then $L(\mathcal{C}) = L(\mathcal{A}) \odot L(\mathcal{B})$ is equivalent to $\bar{\boldsymbol{C}} = \bar{\boldsymbol{A}} \bar{\boldsymbol{B}}$. This implies that the transform $L$ based t-product is equivalent to the matrix-matrix product in the transform domain. Algorithm 1 gives the way for computing t-product.

The t-product enjoys many similar properties to the matrix-matrix product. For example, the t-product is associative, *i.e.*, $\mathcal{A} *_L (\mathcal{B} *_L \mathcal{C}) = (\mathcal{A} *_L \mathcal{B}) *_L \mathcal{C}$.

**Definition 2.2.** ***(Tensor transpose)*** *[9] Let $L$ be any invertible linear transform in (7), and $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$. Then the tensor transpose, denoted as $\mathcal{A}^\top$, satisfies $L(\mathcal{A}^\top)^{(i)} = (L(\mathcal{A})^{(i)})^\top$, $i = 1, \cdots, n_3$.*

**Definition 2.3.** ***(Identity tensor)*** *[9] Let $L$ be any invertible linear transform in (7). Let $\mathcal{I} \in \mathbb{R}^{n \times n \times n_3}$ so that each frontal slice of $L(\mathcal{I}) = \bar{\mathcal{I}}$ is a $n \times n$ sized identity matrix. Then $\mathcal{I} = L^{-1}(\bar{\mathcal{I}})$ is called the identity tensor.*

[1] We restrict $\boldsymbol{L}$ to be a real matrix for the sake of discussions. But our results still hold with simple extensions if necessary for complex $\boldsymbol{L}$ [9].

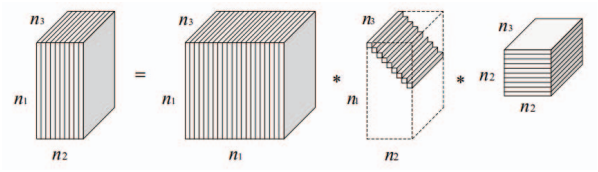


Figure 1: An illustration of the t-SVD of an $n_1 \times n_2 \times n_3$ tensor.

**Algorithm 2** T-SVD based on t-product $*_L$ [9]

**Input:** $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and invertible linear transform $L$.
**Output:** T-SVD components $\boldsymbol{\mathcal{U}}$, $\boldsymbol{\mathcal{S}}$ and $\boldsymbol{\mathcal{V}}$ of $\boldsymbol{\mathcal{A}}$.

1. Compute $\bar{\boldsymbol{\mathcal{A}}} = L(\boldsymbol{\mathcal{A}})$.
2. Compute each frontal slice of $\bar{\boldsymbol{\mathcal{U}}}$, $\bar{\boldsymbol{\mathcal{S}}}$ and $\bar{\boldsymbol{\mathcal{V}}}$ from $\bar{\boldsymbol{\mathcal{A}}}$ by
   **for** $i = 1, \cdots, n_3$ **do**
   $[\bar{\boldsymbol{U}}^{(i)}, \bar{\boldsymbol{S}}^{(i)}, \bar{\boldsymbol{V}}^{(i)}] = \mathrm{SVD}(\bar{\boldsymbol{A}}^{(i)})$;
   **end for**
3. Compute $\boldsymbol{\mathcal{U}} = L^{-1}(\bar{\boldsymbol{\mathcal{U}}})$, $\boldsymbol{\mathcal{S}} = L^{-1}(\bar{\boldsymbol{\mathcal{S}}})$, and $\boldsymbol{\mathcal{V}} = L^{-1}(\bar{\boldsymbol{\mathcal{V}}})$.

It is clear that $\boldsymbol{\mathcal{A}} *_L \boldsymbol{\mathcal{I}} = \boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{I}} *_L \boldsymbol{\mathcal{A}} = \boldsymbol{\mathcal{A}}$ given the appropriate dimensions. The tensor $\bar{\boldsymbol{\mathcal{I}}} = L(\boldsymbol{\mathcal{I}})$ is a tensor with each frontal slice being the identity matrix.

**Definition 2.4.** ***(Orthogonal tensor)*** *[9] Let $L$ be any invertible linear transform in (7). A tensor $\boldsymbol{\mathcal{Q}} \in \mathbb{R}^{n \times n \times n_3}$ is orthogonal if it satisfies $\boldsymbol{\mathcal{Q}}^\top *_L \boldsymbol{\mathcal{Q}} = \boldsymbol{\mathcal{Q}} *_L \boldsymbol{\mathcal{Q}}^\top = \boldsymbol{\mathcal{I}}$.*

**Definition 2.5.** ***(F-diagonal Tensor)*** *[10] A tensor is called f-diagonal if each of its frontal slices is a diagonal matrix.*

**Theorem 2.1.** ***(T-SVD)*** *[9] Let $L$ be any invertible linear transform in (7), and $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$. Then it can be factorized as*

$$\boldsymbol{\mathcal{A}} = \boldsymbol{\mathcal{U}} *_L \boldsymbol{\mathcal{S}} *_L \boldsymbol{\mathcal{V}}^\top, \tag{9}$$

*where $\boldsymbol{\mathcal{U}} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$, $\boldsymbol{\mathcal{V}} \in \mathbb{R}^{n_2 \times n_2 \times n_3}$ are orthogonal, and $\boldsymbol{\mathcal{S}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is an f-diagonal tensor.*

Figure 1 gives an intuitive illustration of the t-SVD factorization. Also, t-SVD can be computed by performing matrix SVD in the transform domain. See Algorithm 2. For any invertible linear transform $L$, we have $L(0) = L^{-1}(0) = 0$. So both $\boldsymbol{\mathcal{S}}$ and $\bar{\boldsymbol{\mathcal{S}}}$ are f-diagonal tensors. We can use their number of nonzero singular tubes to define the tensor tubal rank as in [15].

**Definition 2.6.** ***(Tensor tubal rank)*** *Let $L$ be any invertible linear transform in (7). For $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the tensor tubal rank, denoted as $\mathrm{rank}_t(\boldsymbol{\mathcal{A}})$, is defined as the number of nonzero singular tubes of $\boldsymbol{\mathcal{S}}$, where $\boldsymbol{\mathcal{S}}$ is from the t-SVD of $\boldsymbol{\mathcal{A}} = \boldsymbol{\mathcal{U}} *_L \boldsymbol{\mathcal{S}} *_L \boldsymbol{\mathcal{V}}^\top$. We can write*

$$\mathrm{rank}_t(\boldsymbol{\mathcal{A}}) = \#\{i, \boldsymbol{\mathcal{S}}(i,i,:) \neq 0\}.$$

For $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ with tubal rank $r$, we also have the skinny t-SVD, i.e., $\boldsymbol{\mathcal{A}} = \boldsymbol{\mathcal{U}} *_L \boldsymbol{\mathcal{S}} *_L \boldsymbol{\mathcal{V}}^\top$, where $\boldsymbol{\mathcal{U}} \in \mathbb{R}^{n_1 \times r \times n_3}$, $\boldsymbol{\mathcal{S}} \in \mathbb{R}^{r \times r \times n_3}$, and $\boldsymbol{\mathcal{V}} \in \mathbb{R}^{n_2 \times r \times n_3}$, in which $\boldsymbol{\mathcal{U}}^\top *_L \boldsymbol{\mathcal{U}} = \boldsymbol{\mathcal{I}}$ and $\boldsymbol{\mathcal{V}}^\top *_L \boldsymbol{\mathcal{V}} = \boldsymbol{\mathcal{I}}$. We use the skinny t-SVD throughout this paper. The tensor tubal rank is nonconvex. In Section 3, we will study the low tubal rank tensor completion problem by convex surrogate function minimization. At the following, we show how to define the convex tensor nuclear norm induced by the t-product $*_L$. We can first define the tensor spectral norm as in [15].

**Definition 2.7.** ***(Tensor spectral norm)*** *Let $L$ be any invertible linear transform in (7), and $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$. The tensor spectral norm of $\boldsymbol{\mathcal{A}}$ is defined as $\|\boldsymbol{\mathcal{A}}\| := \|\bar{\boldsymbol{A}}\|$.*

The tensor nuclear norm can be defined as the dual norm of the tensor spectral norm. To this end, we need the following assumption on $L$ given in (7), i.e.,

$$\boldsymbol{L}^\top \boldsymbol{L} = \boldsymbol{L}\boldsymbol{L}^\top = \ell \boldsymbol{I}_{n_3}, \tag{10}$$

where $\ell > 0$ is a constant. Using (10), we have the following important properties,

$$\langle \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}} \rangle = \frac{1}{\ell} \langle \bar{\boldsymbol{A}}, \bar{\boldsymbol{B}} \rangle, \tag{11}$$

$$\|\boldsymbol{\mathcal{A}}\|_F = \frac{1}{\sqrt{\ell}} \|\bar{\boldsymbol{A}}\|_F. \tag{12}$$

For any $\boldsymbol{\mathcal{B}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\tilde{\boldsymbol{B}} \in \mathbb{R}^{n_1 n_3 \times n_2 n_3}$, we have

$$\|\boldsymbol{\mathcal{A}}\|_* := \sup_{\|\boldsymbol{\mathcal{B}}\| \leq 1} \langle \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}} \rangle \tag{13}$$

$$= \sup_{\|\bar{\boldsymbol{B}}\| \leq 1} \frac{1}{\ell} \langle \bar{\boldsymbol{A}}, \bar{\boldsymbol{B}} \rangle \tag{14}$$

$$\leq \frac{1}{\ell} \sup_{\|\tilde{\boldsymbol{B}}\| \leq 1} \langle \bar{\boldsymbol{A}}, \tilde{\boldsymbol{B}} \rangle \tag{15}$$

$$= \frac{1}{\ell} \|\bar{\boldsymbol{A}}\|_*, \tag{16}$$

where (14) uses (11), (15) is due to the fact that $\bar{\boldsymbol{B}}$ is a block diagonal matrix in $\mathbb{R}^{n_1 n_3 \times n_2 n_3}$ while $\tilde{\boldsymbol{B}}$ is an arbitrary matrix in $\mathbb{R}^{n_1 n_3 \times n_2 n_3}$, and (16) uses the fact that the matrix nuclear norm is the dual norm of the matrix spectral norm. On the other hand, let $\boldsymbol{\mathcal{A}} = \boldsymbol{\mathcal{U}} *_L \boldsymbol{\mathcal{S}} *_L \boldsymbol{\mathcal{V}}^\top$ be the t-SVD of $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{B}} = \boldsymbol{\mathcal{U}} *_L \boldsymbol{\mathcal{V}}^\top$. We have

$$\begin{aligned}
\|\boldsymbol{\mathcal{A}}\|_* &= \langle \boldsymbol{\mathcal{U}} *_L \boldsymbol{\mathcal{S}} *_L \boldsymbol{\mathcal{V}}^\top, \boldsymbol{\mathcal{U}} *_L \boldsymbol{\mathcal{V}}^\top \rangle \\
&= \langle \boldsymbol{\mathcal{U}}^\top *_L \boldsymbol{\mathcal{U}} *_L \boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{V}}^\top *_L \boldsymbol{\mathcal{V}} \rangle \\
&= \langle \boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{I}} \rangle \\
&= \frac{1}{\ell} \langle \bar{\boldsymbol{\mathcal{S}}}, \bar{\boldsymbol{I}} \rangle \\
&= \frac{1}{\ell} \mathrm{Tr}(\bar{\boldsymbol{S}}) \\
&= \frac{1}{\ell} \|\bar{\boldsymbol{A}}\|_*.
\end{aligned} \tag{17}$$

Combining (13)-(16) and (17), we then have the following definition of tensor nuclear norm.

**Definition 2.8.** ***(Tensor nuclear norm)*** *Let $L$ be any invertible linear transform in (7) and it satisfies (10), and $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$. The tensor nuclear norm of $\mathcal{A}$ is defined as $\|\mathcal{A}\|_* := \frac{1}{\ell}\|\bar{\boldsymbol{A}}\|_*$.*

If we define the tensor average rank as $\text{rank}_\text{a}(\mathcal{A}) = \frac{1}{\ell}\text{rank}(\bar{\boldsymbol{A}})$, then it can be proved that the above tensor nuclear norm is the convex envelope of the tensor average rank within the domain $\{\mathcal{A} | \|\mathcal{A}\| \leq 1\}$ [15].

## 3. Tensor Completion with Exact Recovery Guarantee

In this section, we consider the low rank tubal tensor completion problem using the linear transform based tensor nuclear norm minimization. Let $L$ be any invertible linear transform in (7) and it satisfies (10). Let $\mathcal{M} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ be an unknown tensor and it has tubal rank $\text{rank}_\text{t}(\mathcal{A}) = r$. Assume that the entries of $\mathcal{M}$ are observed independently with probability $p$. The set of the index of the observed entries is denoted as $\boldsymbol{\Omega}$. In this setting, we denote that $\boldsymbol{\Omega} \sim \text{Ber}(p)$. So, the problem of tensor completion is to recover the underlying low tubal rank tensor $\mathcal{M}$ from the observations $\{\mathcal{M}_{ij}, (i,j,k) \in \boldsymbol{\Omega}\}$. In this work, we solve the tensor completion problem by the following convex program based on the proposed tensor nuclear norm

$$\min_{\mathcal{X}} \|\mathcal{X}\|_*, \text{ s.t. } \mathcal{P}_{\boldsymbol{\Omega}}(\mathcal{X}) = \mathcal{P}_{\boldsymbol{\Omega}}(\mathcal{M}). \tag{18}$$

The above model is convex and thus the optimal solution is computable. Now, the question is, how can we exactly recover $\mathcal{M}$ by solving (18)? It is observed that the recovery is almost impossible for a low-rank matrix which is equal to zero in nearly all of rows or columns [3]. Thus the incoherence conditions are introduced to avoid such pathological situations. For tensor completion, we suffer from the same issue. To avoid the case that $\mathcal{M}$ is too sparse, we need the following standard tensor incoherence conditions

$$\max_{i=1,\cdots,n_1} \|\mathcal{U}^\top *_L \mathring{\mathfrak{e}}_i\|_F \leq \sqrt{\frac{\mu r}{n_1 \ell}}, \tag{19}$$

$$\max_{j=1,\cdots,n_2} \|\mathcal{V}^\top *_L \mathring{\mathfrak{e}}_j\|_F \leq \sqrt{\frac{\mu r}{n_2 \ell}}, \tag{20}$$

where $\mu > 0$, and $\mathring{\mathfrak{e}}_i$ is a tensor basis defined below.

**Definition 3.1.** ***(Standard tensor basis)*** *Let $L$ be any invertible linear transform in (7) and it satisfies (10). We denote $\mathring{\mathfrak{e}}_i$ as the tensor **column basis**, which is a tensor of size $n \times 1 \times n_3$ with the $(i,1,1)$-th entry of $L(\mathring{\mathfrak{e}}_i)$ equaling 1 and the rest equaling 0. Naturally its transpose $\mathring{\mathfrak{e}}_i^\top$ is called **row basis**. The other standard tensor basis is called **tube basis** $\dot{\mathfrak{e}}_k$, which is a tensor of size $1 \times 1 \times n_3$ with the $(1,1,k)$-th entry of $L(\dot{\mathfrak{e}}_k)$ equaling 1 and the rest equaling 0.*

At the following, we denote $n_{(1)} = \max(n_1, n_2)$ and $n_{(2)} = \min(n_1, n_2)$. Then $\ell \leq \mu \leq \frac{n_{(2)}\ell}{r}$.

**Theorem 3.1.** *Let $L$ be any invertible linear transform in (7) and it satisfies (10), and $\mathcal{M} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ with tubal rank $\text{rank}_t(\mathcal{M}) = r$. Let $\mathcal{M} = \mathcal{U} *_L \mathcal{S} *_L \mathcal{V}^\top$ be the skinny t-SVD of $\mathcal{M}$. Suppose that the indices $\boldsymbol{\Omega} \sim Ber(p)$ and the tensor incoherence conditions (19)-(20) hold. There exist universal constants $c_0, c_1, c_2 > 0$ such that if*

$$p \geq \frac{c_0 \mu r \log^2(n_{(1)}\ell)}{n_{(2)}\ell}, \tag{21}$$

*then $\mathcal{M}$ is the unique solution to (18) with probability at least $1 - c_1(n_1+n_2)^{-c_2}$.*

In theory, Theorem 3.1 shows that the optimal solution $\hat{\mathcal{X}}$ to the convex program (18) exactly recovers the underlying low tubal rank tensor $\mathcal{M}$. In particular, to recover a tensor $\mathcal{M} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ with high probability, the sampling complexity is $O(rn_{(1)}n_3 \log^2(n_{(1)}\ell))$. Such a bound is tight compared with $O(rn_{(1)}n_3)$, which is the degree of freedom of the underlying tensor $\mathcal{M}$ with tubal rank $r$. Note that the above result further generalizes the existing low-rank tensor and low-rank matrix completion results. For example, when the discrete Fourier transform matrix is used as the invertible linear transform $L$, the t-product $*_L$ reduces to the t-product in [10], and thus the proposed tensor nuclear norm in Definition 2.8 reduces to the one in [14]. Then the result in Theorem 3.1 is equivalent to the recovery guarantee in [16]. Furthermore, if $\mathcal{M}$ is a matrix, Theorem 3.1 reduces to the recovery guarantee in [5].

Problem (18) can be solved by the standard Alternating Direction Method of Multipliers (ADMM) [12]. We omit the details of ADMM, but give the detail for solving the following key subproblem in ADMM, i.e., for any $\mathcal{Y} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$,

$$\min_{\mathcal{X}} \tau\|\mathcal{X}\|_* + \frac{1}{2}\|\mathcal{X} - \mathcal{Y}\|_F^2. \tag{22}$$

Let $\mathcal{Y} = \mathcal{U} *_L \mathcal{S} *_L \mathcal{V}^\top$ be the tensor-SVD of $\mathcal{Y}$. For any $\tau > 0$, we define the Tensor Singular Value Thresholding (T-SVT) operator as follows

$$\mathcal{D}_\tau(\mathcal{Y}) = \mathcal{U} *_L \mathcal{S}_\tau *_L \mathcal{V}^\top, \tag{23}$$

where

$$\mathcal{S}_\tau = L^{-1}((L(\mathcal{S}) - \tau)_+), \tag{24}$$

where $t_+$ denotes the positive part of $t$, i.e., $t_+ = \max(t, 0)$. This operator simply applies a soft-thresholding rule to $L(\mathcal{S})$, effectively shrinking these towards zero. The T-SVT operator is the proximity operator associated with the proposed tensor nuclear norm.

Table 1: Exact tensor completion on random data. The Discrete Cosine Transform (DCT) is used as the linear transform $L$.

$\boldsymbol{\mathcal{X}}_0 \in \mathbb{R}^{n\times n\times n}, r = \text{rank}_t(\boldsymbol{\mathcal{X}}_0), m = pn^3, d_r = r(2n-r)n$

| $n$ | $r$ | $\frac{m}{d_r}$ | $p$ | $\text{rank}_t(\hat{\boldsymbol{\mathcal{X}}})$ | $\frac{\|\hat{\boldsymbol{\mathcal{X}}}-\boldsymbol{\mathcal{X}}\|_F}{\|\boldsymbol{\mathcal{X}}\|_F}$ |
|---|---|---|---|---|---|
| 50 | 3 | 4 | 0.47 | 3 | 3.5e−6 |
| 50 | 5 | 3 | 0.57 | 5 | 3.1e−6 |
| 50 | 8 | 2 | 0.59 | 8 | 4.2e−6 |
| 50 | 10 | 2 | 0.72 | 10 | 3.1e−6 |
| 100 | 5 | 4 | 0.39 | 5 | 1.1e−5 |
| 100 | 10 | 3 | 0.57 | 10 | 9.9e−6 |
| 100 | 15 | 2 | 0.56 | 15 | 6.8e−6 |
| 100 | 20 | 2 | 0.72 | 20 | 3.8e−6 |
| 200 | 5 | 4 | 0.20 | 5 | 6.2e−5 |
| 200 | 10 | 3 | 0.29 | 10 | 4.8e−5 |
| 200 | 20 | 2 | 0.38 | 20 | 4.7e−5 |
| 200 | 30 | 2 | 0.56 | 30 | 1.9e−5 |

Table 2: Exact tensor completion on random data. A Random Orthogonal Matrix (ROM) is used as the linear transform $L$.

$\boldsymbol{\mathcal{X}}_0 \in \mathbb{R}^{n\times n\times n}, r = \text{rank}_t(\boldsymbol{\mathcal{X}}_0), m = pn^3, d_r = r(2n-r)n$

| $n$ | $r$ | $\frac{m}{d_r}$ | $p$ | $\text{rank}_t(\hat{\boldsymbol{\mathcal{X}}})$ | $\frac{\|\hat{\boldsymbol{\mathcal{X}}}-\boldsymbol{\mathcal{X}}\|_F}{\|\boldsymbol{\mathcal{X}}\|_F}$ |
|---|---|---|---|---|---|
| 50 | 3 | 4 | 0.47 | 3 | 3.6e−6 |
| 50 | 5 | 3 | 0.57 | 5 | 3.0e−6 |
| 50 | 8 | 2 | 0.59 | 8 | 4.4e−6 |
| 50 | 10 | 2 | 0.72 | 10 | 1.3e−6 |
| 100 | 5 | 4 | 0.39 | 5 | 1.5e−5 |
| 100 | 10 | 3 | 0.57 | 10 | 9.7e−6 |
| 100 | 15 | 2 | 0.56 | 15 | 7.0e−6 |
| 100 | 20 | 2 | 0.72 | 20 | 3.9e−6 |
| 200 | 5 | 4 | 0.20 | 5 | 5.8e−5 |
| 200 | 10 | 3 | 0.29 | 10 | 4.6e−5 |
| 200 | 20 | 2 | 0.38 | 20 | 4.7e−5 |
| 200 | 30 | 2 | 0.56 | 30 | 2.0e−5 |

**Theorem 3.2.** *Let $L$ be any invertible linear transform in (7) and it satisfies (10). For any $\tau > 0$ and $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{n_1\times n_2\times n_3}$, the tensor singular value thresholding operator (23) obeys*

$$\mathcal{D}_\tau(\boldsymbol{\mathcal{Y}}) = \arg\min_{\boldsymbol{\mathcal{X}}} \ \tau\|\boldsymbol{\mathcal{X}}\|_* + \frac{1}{2}\|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{Y}}\|_F^2. \tag{25}$$

The main cost of ADMM for solving (18) is to compute $\mathcal{D}_\tau(\boldsymbol{\mathcal{Y}})$ for solving (25). For any general linear transform $L$ in (7), it is easy to see that the per-iteration cost is $O(n_1n_2n_3^2 + n_{(1)}n_{(2)}^2n_3)$. For some specific linear transforms, e.g., DFT, the per-iteration cost can be further reduced. The cost can be further reduced by taking the low-rank structure priori of $\boldsymbol{\mathcal{M}}$ as the matrix case in [12].

## 4. Experiments

In this section, we conduct numerical experiments to corroborate our main results. We first investigate the ability of the convex program (18) to recover tensors with various tubal ranks and sampling rates. We then apply it for image recovery and compare the performance with existing low-rank matrix and low-rank tensor completion models[2].

### 4.1. Exact Recovery on Random Data

We conduct two experiments to demonstrate the practical applicability of the tensor nuclear norm heuristic for recovering low-rank tensors. We first verify the correct recovery guarantee in Theorem 3.1. Note that Theorem 3.1 holds for any invertible linear transform $L$ with $\boldsymbol{L}$ in (7) satisfying (10). We consider two cases of $\boldsymbol{L}$: (1) Discrete Cosine Transform (DCT) [9]; (2) Random Orthogonal Matrix (ROM)[3]. In both cases, (10) holds with $\ell = 1$. We simply consider tensors of size $n \times n \times n$, with $n = [50, 100, 200]$. We generate a tubal rank $r$ tensor by $\boldsymbol{\mathcal{M}} = \boldsymbol{\mathcal{P}} *_L \boldsymbol{\mathcal{Q}}$, where the entries of $\boldsymbol{\mathcal{P}} \in \mathbb{R}^{n\times r\times n}$ and $\boldsymbol{\mathcal{Q}} \in \mathbb{R}^{r\times n\times n}$ are independently sampled from an $\mathcal{N}(0, 1/n)$ distribution. Then we sample $m = pn^3$ elements uniformly from $\boldsymbol{\mathcal{M}}$ to form the known entries. A useful quantity for reference is $d_r = r(2n-r)n$. With $\mathcal{P}_{\boldsymbol{\Omega}}(\boldsymbol{\mathcal{M}})$, we solve (18) and obtain the solution $\hat{\boldsymbol{\mathcal{X}}}$. Then we report the tubal rank of $\hat{\boldsymbol{\mathcal{X}}}$ and the relative errors $\|\hat{\boldsymbol{\mathcal{X}}} - \boldsymbol{\mathcal{M}}\|_F/\|\boldsymbol{\mathcal{M}}\|_F$. Table 1 and 2 respectively give the recovery results for the two different choices of the linear transforms $L$. It can be seen that the convex program (18) gives the correct rank estimation of $\boldsymbol{\mathcal{M}}$ in all cases and the relative errors are small. Thus, these results well verify the correct recovery guarantee as claimed in Theorem 3.1 for convex program (18).

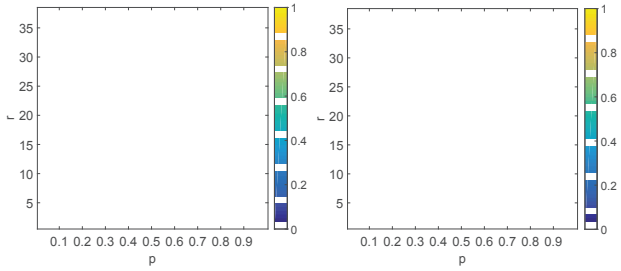
Theorem 3.1 shows the perfect recovery for incoherent tensor based on $\text{rank}_t(\boldsymbol{\mathcal{M}})$ and the sampling rate $p$. Now we examine the recovery phenomenon with varying tubal rank of $\boldsymbol{\mathcal{M}}$ and varying sampling rate $p$. We generate $\boldsymbol{\mathcal{M}} \in \mathbb{R}^{n\times n\times n}$, where $n = 50$, in the same way as the above experiment. Then we sample $m = pn^3$ elements uniformly to form the known entries. We set $p = [0.01 : 0.01 : 0.99]$, and $r = [1, 2, \ldots, 38]$. For each $(r, p)$ pair, we simulate 10 test instances and declare a trial to be successful if the recovered $\hat{\boldsymbol{\mathcal{X}}}$ satisfies $\|\hat{\boldsymbol{\mathcal{X}}} - \boldsymbol{\mathcal{M}}\|_F/\|\boldsymbol{\mathcal{M}}\|_F \le 10^{-3}$. Figure 2 plots the fraction of correct recovery for each pair (blue = 0% and yellow = 100%) for different choices of the linear transform $L$. It can be seen that the recovery phenomenons in both cases are very similar. Both have a large region in which the recovery is correct. This verifies our main result in Theorem 3.1 that the recovery guarantee is invariant to different choices of the linear transforms when property (10) holds. Our results are also consistent with the phenomenons in existing low-rank matrix completion [3, 5] and low-rank tensor completion [16].

### 4.2. Tensor Completion for Image Recovery

We consider the application of tensor completion for image recovery. The motivation is that the real color images

[2]Codes of our method: http://github.com/canyilu.

[3]Codes: https://www.mathworks.com/matlabcentral/fileexchange/11783-randorthmat.




(a) discrete cosine transform (b) random orthogonal matrix

Figure 2: Phase transitions for tensor completion. Fraction of correct recoveries across 10 trials, as a function of sampling rate $p$ (x-axis) and tubal rank $\text{rank}_t(\boldsymbol{\mathcal{M}})$ (y-axis). Left: discrete cosine transform is used as the linear transform $L$. Right: random orthogonal matrix is used as the linear transform $L$.

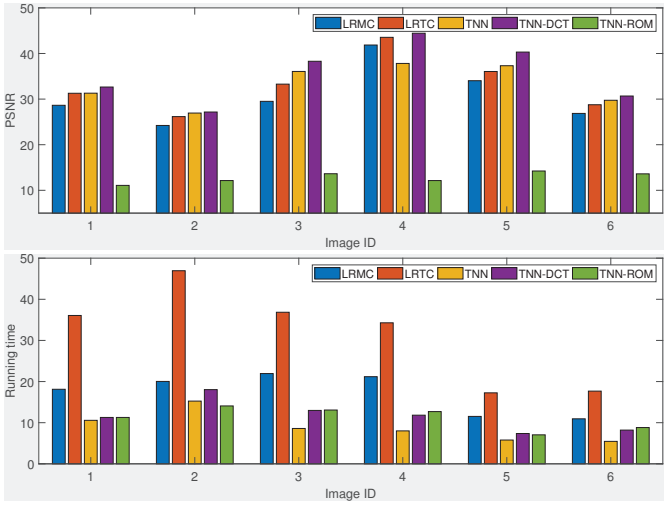



Figure 3: Comparison of the PSNR values and running time (seconds) obtained by LRMC, LRTC, TNN, TNN-DCT and TNN-ROM on the tested images in Figure 4.

can be well approximated by low-rank matrices on the three channels independently. If we treat a color image as a three way tensor, then it can be well approximated by low rank tensors. There have many works which use such observations, e.g., [13, 28, 14, 15, 7], for image recovery.

Note that the t-product induced tensor nuclear norm is orientation dependent. The recovery performance may be different for different tensors constructed from a color image. For a color image with size $h \times w$, we can construct a tensor $\boldsymbol{\mathcal{M}} \in \mathbb{R}^{h\times 3\times w}$, where the lateral slices correspond to the three channels[4]. We randomly set $m = 3phw$ entries to be observed, where we use $p = 0.3$ in this experiment. We consider four methods for image recovery and compare their performance: (1) LRMC [3]: low-rank matrix completion applied on each channel of images separably; (2) LRTC [13]: low-rank tensor completion model in (2); (3) TNN [16]: tensor nuclear norm (TNN) based tensor completion model (a special case of our model (18) by using the discrete Fourier transform as the linear transform $L$); (4) TNN-DCT: our TNN minimization model (18) by using the Discrete Cosine Transform (DCT)[5] as the linear transform $L$; (5) TNN-ROM: our TNN minimization model (18) by using the Random Orthogonal Matrix (ROM) as the linear transform $L$. The Peak Signal-to-Noise Ratio (PSNR) [16] is used to evaluate the recovery performance.

Figure 4 shows the recovered images obtained by different methods on some sample images. Figure 3 further shows the comparison on the PSNR values and running time on the tested images. From these results, it can be seen that our TNN-DCT achieves the best performance in most cases, while TNN-ROM achieves the worst performance. In particular, TNN-DCT has better recovery performance than TNN on the edges of images as shown in Figure 4. This implies that the discrete Fourier transform used in [10] may not be the best in image analysis, though the reason is not very clear now. However, if the random orthogonal matrix is used as the linear transform $L$, the results are very bad. It may not be a good choice in image analysis. In a summary, we observe that the choice of the linear transform $L$ is crucial in practice, though the best linear transform is currently not clear. For the efficiency, Figure 3 shows that our new methods is comparable with RPCA and the existing TNN method.

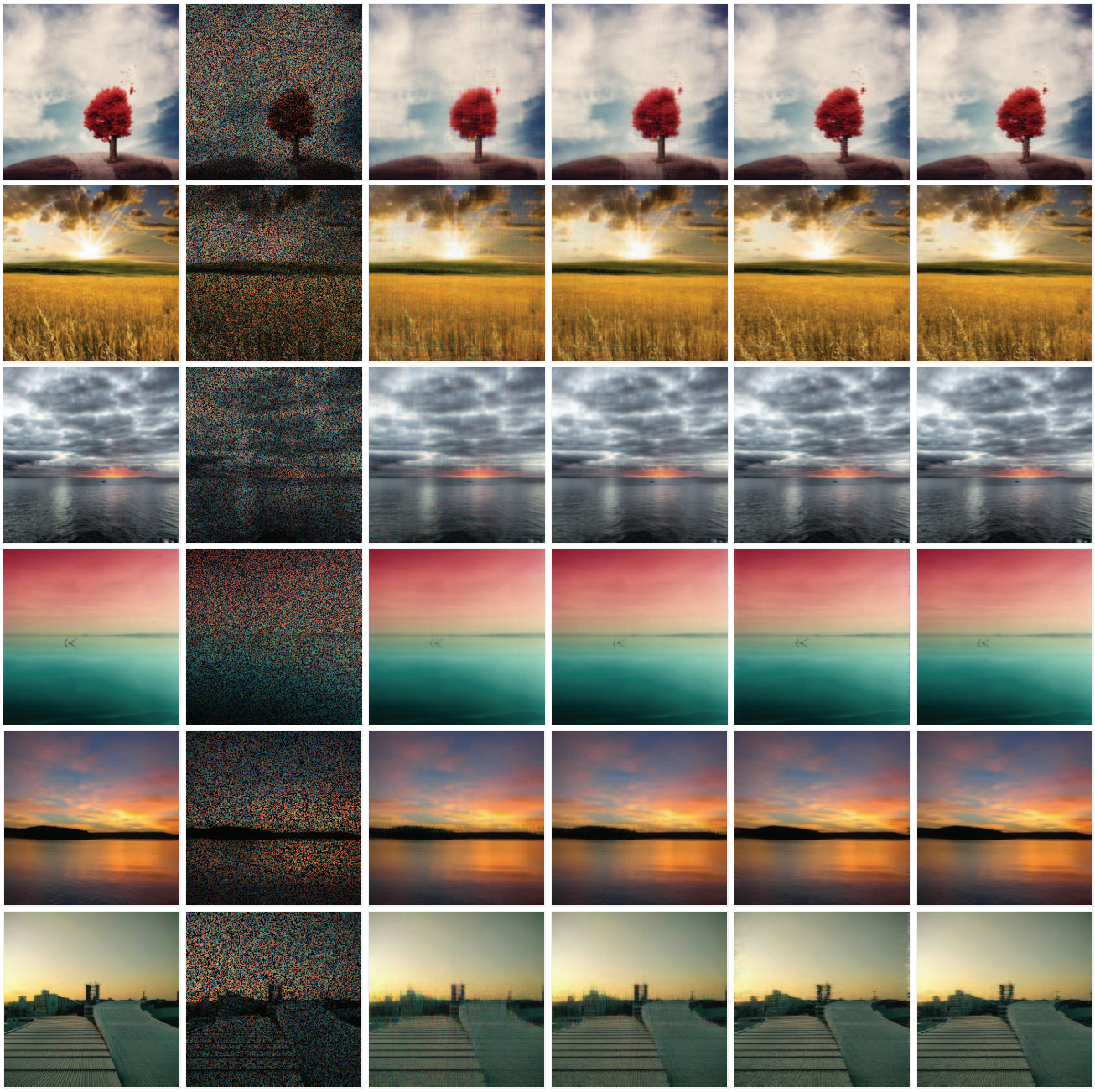
## 5. Conclusions

Inspired by the recent invertible linear transforms based t-product [9] which generalizes the existing discrete Fourier transform based t-product [10], we give more general definitions of tensor tubal rank and tensor nuclear norm. We show that when the invertible linear transforms satisfy certain conditions, the tensor completion problem can be solved by the proposed convex tensor nuclear norm minimization. In theory, we give the order optimal bound for the guarantee of exact recovery. It is interesting that the main result holds for a broad choices of the invertible linear transforms. This greatly generalizes the existing result for low tubal rank tensor recovery. Numerical experiments verified our theory and the application on image recovery demonstrates the effectiveness of our model.

There have some interesting future works. First, as the t-product $*_L$ and our tensor nuclear norm depend on the used linear transform, it is interesting to learn the linear transform from the data in different tasks. It is expected to improve the learning performance by optimizing the linear transfor-

[4]It is observed in [16] that this way of tensor construction achieves the best performance in practice.

[5]DCT is another interesting invertible linear transform discussed in the original work of the new t-product definition in [9].



(a) orignal image (b) observed image (c) LRMC (d) LRTC (e) TNN (f) TNN-DCT

Figure 4: Performance comparison for image recovery on some sample images. (a) Original image; (b) observed image; (c)-(f) recovered images by LRMC, LRTC, TNN and TNN-DCT, respectively. **Best viewed in $\times 2$ sized color pdf file.**

m for different types of data. Second, beyond the tensor completion problem studied in this work, it is interesting to use the tensor nuclear norm for some other tasks, e.g., tensor robust PCA [14] and low-rank representation models [17]. Also, it is always of great interest to apply our new method for different applications, e.g., motion segmentation and background modeling [4].

## Acknowledgement

Xi Peng was supported by the Fundamental Research Funds for the Central Universities under Grant YJ201748, by NFSC under Grant 61806135, by NFSC for Distinguished Young Scholar under Grant 61625204, and by the Key Program of NFSC under Grant 61836006.

## References


[1] A. Anandkumar, R. Ge, D. Hsu, S. M. Kakade, and M. Telgarsky. Tensor decompositions for learning latent variable models. *J. Machine Learning Research*, 15(1):2773–2832, 2014. 1

[2] R. S. Cabral, F. Torre, J. P. Costeira, and A. Bernardino. Matrix completion for multi-label image classification. In *Advances in Neural Information Processing Systems*, pages 190–198, 2011. 1

[3] E. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717–772, 2009. 1, 5, 6, 7

[4] E. J. Candès, X. D. Li, Y. Ma, and J. Wright. Robust principal component analysis? *J. ACM*, 58(3), 2011. 8

[5] Y. Chen. Incoherence-optimal matrix completion. *IEEE Trans. Information Theory*, 61(5):2909–2923, May 2015. 1, 2, 5, 6

[6] L. De Lathauwer and B. De Moor. From matrix to tensor: Multilinear algebra and signal processing. In *Institute of Mathematics and its Applications Conference Series*, volume 67, pages 1–16. Citeseer, 1998. 1

[7] S. Gandy, B. Recht, and I. Yamada. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems*, 27(2):025010, 2011. 2, 7

[8] D. F. Gleich, C. Greif, and J. M. Varah. The power and arnoldi methods in an algebra of circulants. *Numerical Linear Algebra with Applications*, 20(5):809–831, 2013. 2

[9] E. Kernfeld, M. Kilmer, and S. Aeron. Tensor–tensor products with invertible linear transforms. *Linear Algebra and its Applications*, 485:545–570, 2015. 1, 2, 3, 4, 6, 7

[10] M. E. Kilmer and C. D. Martin. Factorization strategies for third-order tensors. *Linear Algebra and its Applications*, 435(3):641–658, 2011. 2, 3, 4, 5, 7

[11] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Rev.*, 51(3):455–500, 2009. 1, 2

[12] Z. Lin, M. Chen, and Y. Ma. The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1009.5055*, 2010. 5, 6

[13] J. Liu, P. Musialski, P. Wonka, and J. Ye. Tensor completion for estimating missing values in visual data. *IEEE Trans. Pattern Recognition and Machine Intelligence*, 35(1):208–220, 2013. 1, 2, 7

[14] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan. Tensor robust principal component analysis: Exact recovery of corrupted low-rank tensors via convex optimization. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*. IEEE, 2016. 2, 5, 7, 8

[15] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan. Tensor robust principal component analysis with a new tensor nuclear norm. *IEEE Trans. Pattern Recognition and Machine Intelligence*, 2018. 2, 4, 5, 7

[16] C. Lu, J. Feng, Z. Lin, and S. Yan. Exact low tubal rank tensor recovery from gaussian measurements. In *Proc. Int'l Joint Conf. Artificial Intelligence*, 2018. 1, 2, 5, 6, 7

[17] C. Lu, Z. Lin, and S. Yan. Smoothed low rank and sparse matrix recovery by iteratively reweighted least squares minimization. *IEEE Trans. Image Processing*, 24(2):646–654, Feb 2015. 8

[18] C. Lu, J. Tang, S. Yan, and Z. Lin. Nonconvex nonsmooth low rank minimization via iteratively reweighted nuclear norm. *IEEE Trans. Image Processing*, 25(2):829–839, 2016. 1

[19] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. IEEE Int'l Conf. Computer Vision*, volume 2, pages 416–423. IEEE, 2001.

[20] N. Mesgarani, M. Slaney, S. Shamma, et al. Discrimination of speech from nonspeech based on multiscale spectro-temporal modulations. *IEEE Trans. Audio, Speech, and Language Processing*, 14(3):920–930, 2006. 1

[21] C. Mu, B. Huang, J. Wright, and D. Goldfarb. Square deal: Lower bounds and improved relaxations for tensor recovery. In *Proc. Int'l Conf. Machine Learning*, pages 73–81, 2014. 2

[22] B. Romera-Paredes and M. Pontil. A new convex relaxation for tensor completion. In *Advances in Neural Information Processing Systems*, pages 2967–2975, 2013. 2

[23] B. Savas and L. Eldén. Handwritten digit classification using higher order singular value decomposition. *Pattern recognition*, 40(3):993–1003, 2007. 1

[24] M. Signoretto, Q. T. Dinh, L. De Lathauwer, and J. A. Suykens. Learning with tensors: a framework based on convex optimization and spectral regularization. *Machine Learning*, 94(3):303–351, 2014. 2

[25] R. Tomioka, K. Hayashi, and H. Kashima. Estimation of low-rank tensors via convex optimization. *arXiv preprint arXiv:1010.0789*, 2010. 2

[26] L. R. Tucker. Implications of factor analysis of three-way matrices for measurement of change. *Problems in measuring change*, 15, 1963. 1

[27] M. A. O. Vasilescu and D. Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In *Proc. European Conf. Computer Vision*, pages 447–460. Springer, 2002. 1

[28] Y. Xu, R. Hao, W. Yin, and Z. Su. Parallel matrix factorization for low-rank tensor completion. *Inverse Problems and Imaging*, 9(2):601–624, 2015. 7