Efficient and Learnable Transformed Tensor Nuclear Norm with Exact Recoverable Theory

Anonymous Author(s)
Affiliation
Address
email

Abstract

The tensor nuclear norm represents the low-rank property of tensor slices under a transformation. Finding a good transformation is crucial for the tensor nuclear norm. However, existing transformations are either fixed and not adaptable to the data, leading to ineffective results, or they are nonlinear and non-invertible, which prevents theoretical guarantees for the transformed tensor nuclear norm. Besides, some transformations are too complex and computationally expensive. To address these issues, this paper first proposes a fast data-adaptive and learnable column-orthogonal transformation learning framework with an exact recoverable theoretical guarantee. Extensive experiments have validated the effectiveness of the proposed models and theories.

1 Introduction

In real-life scenarios, many high-dimensional tensor data, such as hyperspectral images (HSIs), multispectral images (MSIs), and multi-frame videos, exhibit strong low-rank properties. Leveraging such low-rank structures of tensor data is crucial for solving tensor data restoration tasks, including but not limited to tensor completion (TC) [1, 2] and tensor robust principal component analysis (TRPCA) [3, 4]. Numerous methods have achieved outstanding results in practical applications by exploiting the low-rank property of tensors, such as video processing [5, 6], hyperspectral denoising [7, 8, 9], classification [10, 11].

There are various definitions of tensor rank, which differ from the rank used for matrices [12, 1]. Two well-known types of tensor decomposition are based on the CANDECOMP/PARAFAC (CP) and Tucker decompositions, which define the CP rank and Tucker rank, respectively [12]. These decompositions have been widely studied and have demonstrated competitive performance in low-rank tensor recovery. Computing the CP rank is known to be NP-hard, and a clear convex surrogate for this rank has not been established. On the other hand, computing the Tucker rank involves unfolding tensors along each mode into matrices, which may result in the loss of intrinsic high-order interactive information. In addition to these two ranks, the tensor tubal rank is also commonly used for tensor decomposition [13]. This rank is computed via tensor singular value decomposition (t-SVD), which was initially derived from a novel definition of the tensor-tensor (t-t) product [14]. Unlike other methods, t-SVD can operate on an integral third-order tensor without reshaping it into matrices, by using the discrete Fourier transform (DFT). For a third-order tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, assuming that its third mode has a low-rank property, the transformed tensor $\overline{\boldsymbol{\mathcal{A}}}$ can be obtained as follows:

$$\overline{\boldsymbol{\mathcal{A}}} = \boldsymbol{\mathcal{A}} \times_3 \mathbf{L}, \tag{1}$$

where $\times_3$ denotes mode-3 tensor product [12], and $\mathbf{L} \in \mathbb{R}^{n_3 \times n_3}$ is corresponding DFT matrix which satisfies $\mathbf{L}\mathbf{L}^T = \mathbf{L}^T\mathbf{L} = n_3\boldsymbol{I}$. Then the definition of the tensor tubal rank of $\boldsymbol{\mathcal{A}}$ is $\text{rank}_t(\boldsymbol{\mathcal{A}}) =$

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

Table 1: The characteristics of different transformed TNN.

Methods	TNN [2]	DCTNN [28]	UTNN [30]	WTNN [29]	CTNN [32]	FTNN [31]	S2NTNN [23]	Q-rank [24]	SALTS [25]	Ours
Transform	FFT	DCT	Unitary	Wavelet	Couple	Framelet	DNN	Unitary	Unitary	COM
Learnable?	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓
Theory?	✓	✓	✓	✓	✗	✗	✗	✓	✗	✓
Speed	Moderate	Moderate	Moderate	Moderate	Slow	Slow	Fast	Very slow	Very slow	**Fast**

$\sum_{i=1}^{n_3}$ rank$(\overline{\mathcal{A}}(:,:,i))$, where $\overline{\mathcal{A}}(:,:,i)$ is the frontal slice of $\overline{\mathcal{A}}$. Since the minimization of the tubal rank is an NP-hard problem. Zhang et al. [15] built a convex surrogate of the tensor tubal rank, named the tensor nuclear norm (**TNN**) by summing the matrix nuclear norm of each frontal slice under DFT. Thus the DFT-transformed TNN is defined as:

$$\|\mathcal{A}\|_* = \sum_{i=1}^{n_3} \|\overline{\mathcal{A}}(:,:,i)\|_* = \sum_{i=1}^{n_3} \|\overline{\boldsymbol{A}}^{(k)}\|_*. \tag{2}$$

Based on the DFT transformed TNN, Zhang and Aeron [2] and Lu et al. [3] give the exact recovery theorem for TC and TRPCA task by minimizing the TNN norm, respectively. Since then, many variants of DFT transformed TNN are proposed, such as weight TNN [16], partial sum of TNN (PSTNN) [17], Schatten-p norm TNN [18], p-shrinkage TNN [19], and many others [20, 21, 22].

Referring to Eq. (1), if we substitute the DFT matrix with another transform matrix/operator $\mathbf{L}$, we can obtain a transformed tensor and corresponding induced TNN norms that differ from those obtained using DFT. Hence, a crucial question arises: what type of transform matrix/operator is appropriate? Intuitively, a suitable transform operator should satisfy the following three criteria:

1) **Data adaptation.** The design of transform operators must depend on the data to better utilize its characteristics, which is a recent viewpoint. Works such as S2NTNN [23], Q-rank [24], and SALTS [25] have employed various methods to learn transform matrices from data. S2NTNN uses deep neural networks, Q-rank introduces a new algebraic definition, and SALTS uses SVD decomposition. Although only Q-rank has theoretical guarantees, updating the transform matrix and tensor recovery are independent processes that take a long time, making it impractical for real-world tasks.
2) **Theoretical guarantee** Theoretical guarantees are crucial for both models and algorithms. Currently, the exact recoverable guarantees are based on fixed linear invertible transforms, such as DFT, discrete cosine transform (DCT) [26, 27, 28], wavelet transformation [29], and unitary transformation [30], but they lack adaptability to data. In addition, there are fixed complex transforms that do not have recoverable theoretical guarantees, such as framelet transform [31], and coupe transform [32].
3) **Good Performance** Good transforms should improve restoration performance.

To achieve these objectives, this paper leverages the tensor structure and exploits the low-rank property of the third mode of the tensor to learn an adaptive column-orthogonal matrix (COM) transform for each data instance. Specifically, we model the low-rank tensor to be restored as the product of a smaller-sized factor tensor and a COM. This modeling approach effectively captures the low-rank structure of the tensor and facilitates the learning of the COM transform. Moreover, due to the reduced size of the factor tensor compared to the original tensor, our proposed model achieves accelerated computation. Additionally, we provide theoretical guarantees for the recoverability of our proposed model. To facilitate comparison, we present some classical transform-based tensor nuclear norm (TNN) approaches in Table 1. It can be observed from the table that only our modeling approach can stand out by simultaneously considering data adaptability, theoretical guarantees, and computational efficiency. In summary, this article first presents an efficient learnable transformed tensor nuclear norm (TNN) model with recoverable theoretical guarantees.

2 Notations and Preliminaries

2.1 Notations

In this paper, we denote tensors by boldface Euler script letters, e.g., $\mathcal{A}$. Matrices are denoted by boldface capital letters, e.g., $\boldsymbol{A}$; vectors are denoted by boldface lowercase letters, e.g., $\boldsymbol{a}$, and scalars are denoted by lowercase letters, e.g., a. We denote $\boldsymbol{I}_n$ as the $n \times n$ identity matrix. For a 3-order tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the frontal slice $\mathcal{A}(:,:,i)$ is denoted compactly as $\boldsymbol{A}^{(i)}$. The tube is denoted as $\mathcal{A}(i,j,:)$. The mode-n unfolding matrix of $\mathcal{A}$ is denoted as $\mathbf{A}_{(n)} = \text{unfold}_n(\mathcal{A})$, and $\text{fold}_n(\mathbf{A}_{(n)}) = \mathcal{A}$, where fold_n is the inverse of unfolding operator. The mode-n product of a tensor $\mathcal{X} \in \mathbb{R}^{I_1\times I_2\times I_3}$ and a matrix $\mathbf{A} \in \mathbb{R}^{J_n\times I_n}$ is denoted as $\mathcal{Y} := \mathcal{X} \times_n \mathbf{A}$ (see definition in [12]). Some norms of vector, matrix and tensor are used. We denote the $\|\mathcal{A}\|_1 = \sum_{ijk}|a_{ijk}|$, the infinity norm as $\|\mathcal{A}\|_\infty = \max_{ijk}|a_{ijk}|$ and the Frobenius norm as $\|\mathcal{A}\|_F = \sqrt{\sum_{ijk}|a_{ijk}|^2}$, respectively.

2.2 Adaptive Transformation

For a third-order tensor $\mathcal{A} \in \mathbb{R}^{n_1\times n_2\times n_3}$, assuming that its third mode has low-rank property, it can be factorized as

$$\mathcal{A} = \mathcal{U} \times_3 \mathbf{V}, \tag{3}$$

where $\times_3$ denotes mode-3 tensor product, $\mathcal{U} \in \mathbb{R}^{n_1\times n_2\times r_3}$, $\mathbf{V} \in \mathbb{R}^{n_3\times r_3}(r_3 \le n_3)$ satisfying $\mathbf{V}^T\mathbf{V} = \boldsymbol{I}$ and $r_3 = \text{Rank}(\mathbf{A}_{(3)})$. According to low-rank tensor decomposition (3), we have.

$$\mathcal{U} = \mathcal{A} \times_3 \mathbf{V}^T \iff \mathbf{U}_{(3)} = \mathbf{U}_{(3)}\mathbf{V}^T\mathbf{V} = \mathbf{A}_{(3)}\mathbf{V}. \tag{4}$$

Therefore, if we regard $\mathcal{U}$ as a transformed tensor $\overline{\mathcal{A}}$, then $\mathbf{V}^T$ can be regarded as the transform matrix $\mathbf{L}$, and $\mathbf{V}$ is the inverse transform of $\mathbf{V}^T$. Then we denote the TNN under the COM learned from the data as the Adaptive TNN (**ATNN**), which can be reformulated as:

$$\|\overline{\mathcal{A}}\|_* = \sum_{k=1}^{r_3}\|\overline{\mathbf{A}}^{(k)}\|_* = \sum_{k=1}^{R}\|(\mathcal{A}\times_3\mathbf{L}^T)^{(k)}\|_*,\ \text{s.t.}\ \mathcal{A} = \mathcal{A}\times_3\mathbf{L}^T\times_3\mathbf{L}. \tag{5}$$

Remark 1 *It should be noted that comparing Eq. (5) and Eq. (2), it can be seen that ATNN has faster solution efficiency than DFT-transformed TNN since the transformed tensor under COM transform has fewer slices. The stronger the low rank of the tensor, that is, the lower the r_3/n_3 value, the higher the solution efficiency of ATNN can be obtained. However, since we want to ensure that the information of $\mathcal{A}$ with a rank of Rank$(\mathbf{A}_{(3)})$ before and after the transform will not be lost, i.e., $\mathcal{A} = \mathcal{A}\times_3\mathbf{L}^T\times_3\mathbf{L}$ is established, the condition $r_3 \ge$ Rank$(\mathbf{A}_{(3)})$ must hold.*

2.3 T-product and T-SVD

Here, we give the definitions of t-product and t-SVD based on COM transform.

For $\mathcal{A} \in \mathbb{R}^{n_1\times n_2\times n_3}, \mathcal{B} \in \mathbb{R}^{n_2\times n_4\times n_3}$, the COM $\mathbf{L}^T$ transformed tensor of $\mathcal{A}, \mathcal{B}$ are $\overline{\mathcal{A}} = \mathcal{A}\times\mathbf{L}^T \in \mathbb{R}^{n_1\times n_2\times R}, \overline{\mathcal{B}} = \mathcal{B}\times\mathbf{L}^T \in \mathbb{R}^{n_2\times n_4\times R}$, respectively, via Eq. (1), then we define

$$\overline{A} = \text{bdiag}(\overline{\mathcal{A}}) = \begin{bmatrix} \overline{\mathbf{A}}^{(1)} & & & \\ & \overline{\mathbf{A}}^{(2)} & & \\ & & \ddots & \\ & & & \overline{\mathbf{A}}^{(R)} \end{bmatrix}, \overline{\mathcal{A}} = \text{bfold}\left(\overline{A}\right). \tag{6}$$

Definition 1 (T-product) *Let $\mathcal{A} \in \mathbb{R}^{n_1\times n_2\times n_3}, \mathcal{B} \in \mathbb{R}^{n_2\times n_4\times n_3}$ and COM $\mathbf{L}^T \in \mathbb{R}^{r_3\times n_3}, (r_3 \le n_3)$ satisfying $\mathbf{L}^T\mathbf{L} = \boldsymbol{I}_R$, then the t-product under transform $\mathbf{L}^T$ is defined as*

$$\mathcal{C} = \mathcal{A} *_L \mathcal{B} = \mathit{bfold}(\mathit{bdiag}(\overline{\mathcal{A}})\mathit{bdiag}(\overline{\mathcal{B}})) \times_3 \mathbf{L} = \mathit{bfold}(\overline{A}\,\overline{B}) \times_3 \mathbf{L} \in \mathbb{R}^{n1\times n_4\times n_3}, \tag{7}$$

where $\overline{\mathcal{A}} = \mathcal{A}\times_3\mathbf{L}^T \in \mathbb{R}^{n_1\times n_2\times r_3}$ and $\overline{\mathcal{B}} = \mathcal{B}\times_3\mathbf{L}^T \in \mathbb{R}^{n_2\times n_4\times r_3}$.

According to the Definition 1, we have $\mathcal{C} = \mathcal{A} *_L \mathcal{B} \iff \overline{C} = \overline{A}\,\overline{B}$ since $\text{bfold}(\overline{\mathbf{C}}) = \overline{\mathcal{C}} = \mathcal{C}\times_3\mathbf{L}^T = \text{bfold}(\overline{A}\,\overline{B})\times_3\mathbf{L}\times_3\mathbf{L}^T = \text{bfold}(\overline{A}\,\overline{B})\times_3(\mathbf{L}^T\mathbf{L}) = \text{bfold}(\overline{A}\,\overline{B})$.

The t-product enjoys many similar properties to the matrix-matrix product. For example, the t-product is associate, i.e., $\mathcal{A} * (\mathcal{B} * \mathcal{C}) = (\mathcal{A} * \mathcal{B}) * \mathcal{C}$. We also need some other concepts on tensors.

Definition 2 (Transpose) *The transpose of a tensor $\mathcal{A} \in \mathbb{R}^{n_1\times n_2\times n_3}$ is the tensor $\mathcal{A}^T \in \mathbb{R}^{n_2\times n_1\times n_3}$ obtained by transposing each of the frontal slices.*

Definition 3 (Identity tensor) *A third-order tensor* $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{n \times n \times n_3}$ *is called identity tensor if it satisfies that each frontal slice is identity matrix, i.e.,* $\mathbf{A}^{(i)} = \boldsymbol{I}$ *for all* $i = 1, \cdots, n_3$.

Definition 4 (Orthogonal tensor) *A third-order tensor* $\boldsymbol{\mathcal{Q}} \in \mathbb{R}^{n \times n \times n_3}$ *is called orthogonal tensor if it satisfies that* $\boldsymbol{\mathcal{Q}}^T *_L \boldsymbol{\mathcal{Q}} = \boldsymbol{\mathcal{Q}} *_L \boldsymbol{\mathcal{Q}}^T = \boldsymbol{\mathcal{I}}$.

Definition 5 (F-diagonal tensor) *A tensor is called f-diagonal if each of its frontal slices is a diagonal matrix.*

Theorem 1 (T-SVD) *Let* $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$. *Then it can be factorized as*

$$\boldsymbol{\mathcal{A}} = \boldsymbol{\mathcal{U}} *_L \boldsymbol{\mathcal{S}} *_L \boldsymbol{\mathcal{V}}^T, \tag{8}$$

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

By replacing DFT transform with COM transform $\mathbf{L}^T$, we can prove the above Theorem [3].

Definition 6 (Tensor tubal rank [14] & TNN [3]) *For* $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, *the tensor tubal rank, denoted as 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* $\boldsymbol{\mathcal{A}} = \boldsymbol{\mathcal{U}} *_L \boldsymbol{\mathcal{S}} *_L \boldsymbol{\mathcal{V}}^T$. *We can write*

$$\textit{rank}_t(\boldsymbol{\mathcal{A}}) = \#\{i, \boldsymbol{\mathcal{S}}(i,i,:) \neq \mathbf{0}\}. \tag{9}$$

And its tensor nuclear norm (TNN) is defined as

$$\|\boldsymbol{\mathcal{A}}\|_* = \sum_i \|\boldsymbol{\mathcal{S}}(i,i,:)\|_1 = \|\boldsymbol{\mathcal{S}}\|_1. \tag{10}$$

Using the t-product definition, we can get $\boldsymbol{\mathcal{A}} = \boldsymbol{\mathcal{U}} *_L \boldsymbol{\mathcal{S}} *_L \boldsymbol{\mathcal{V}}^T \iff \overline{\mathbf{A}} = \overline{\mathbf{U}}\,\overline{\mathbf{S}}\,\overline{\mathbf{V}^T}$, thus we have

$$\|\boldsymbol{\mathcal{A}}\|_* = \|\boldsymbol{\mathcal{S}}\|_1 = \|\overline{\boldsymbol{\mathcal{S}}}\|_* = \|\overline{\boldsymbol{A}}\|_* = \|\overline{\boldsymbol{\mathcal{A}}}\|_* \tag{11}$$

by combing Eq. (5), Eq. (6) and Eq. (10).

3 Tensor Recovery via ATNN Minimization

3.1 Models

The observed tensor and the tensor that needs to be recovered are denoted as $\boldsymbol{\mathcal{Y}}$ and $\boldsymbol{\mathcal{X}}_0$, respectively. For the tensor completion (TC), the observation $\boldsymbol{\mathcal{Y}}$ has the support set $\boldsymbol{\Omega} \sim \text{Ber}(\rho)$, i.e., $\boldsymbol{\mathcal{P}}_{\boldsymbol{\Omega}}(\boldsymbol{\mathcal{Y}}) = \boldsymbol{\mathcal{P}}_{\boldsymbol{\Omega}}(\boldsymbol{\mathcal{X}}_0)$. For the tensor robust principal component analysis (TRPCA), the observation $\boldsymbol{\mathcal{Y}}$ is corrupted with a sparse component $\boldsymbol{\mathcal{E}}_0$ (which may represent foreground and sparse noise), denoted as $\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{X}}_0 + \boldsymbol{\mathcal{E}}_0$.

If the COM $\mathbf{L}^T$ satisfying Eq. (5) is known, we can obtain the following two models:

$$\begin{aligned}
&(\text{TRPCA}): \max_{\boldsymbol{\mathcal{X}},\boldsymbol{\mathcal{S}}} \ \|\boldsymbol{\mathcal{X}} \times_3 \mathbf{L}^T\|_* + \lambda\|\boldsymbol{\mathcal{S}}\|_1, \ s.t.\ \boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{X}} + \boldsymbol{\mathcal{E}}, \\
&(\text{TC}): \max_{\boldsymbol{\mathcal{X}}} \ \|\boldsymbol{\mathcal{X}} \times_3 \mathbf{L}^T\|_*, \ s.t.\ \boldsymbol{\mathcal{P}}_{\boldsymbol{\Omega}}(\boldsymbol{\mathcal{Y}}) = \boldsymbol{\mathcal{P}}_{\boldsymbol{\Omega}}(\boldsymbol{\mathcal{X}}).
\end{aligned} \tag{12}$$

Actually, it is often not possible to obtain $\mathbf{L}^T$ that satisfies Eq. (5) in advance. Recall Eq. (5), where the constraint $\boldsymbol{\mathcal{A}} = \boldsymbol{\mathcal{A}} \times_3 \mathbf{L}^T \times_3 \mathbf{L}$ shows that the information of $\boldsymbol{\mathcal{A}}$ after the change and inverse change will not be lost, as long as $\mathbf{L}$ is obtained from the SVD decomposition of $\boldsymbol{\mathcal{X}}$, Eq. (5) can be satisfied. Hence, we can learn a suitable COM $\mathbf{L}$ from the data. By decomposing $\boldsymbol{\mathcal{X}}$ as $\boldsymbol{\mathcal{X}} = \overline{\boldsymbol{\mathcal{M}}} \times_3 \mathbf{L}$ and setting $\boldsymbol{\mathcal{M}} = \boldsymbol{\mathcal{X}} \times_3 \mathbf{L}^T$, we can obtain the following alternative model to Eq. (12):

$$\begin{aligned}
&(\text{TRPCA}): \max_{\overline{\boldsymbol{\mathcal{M}}},\boldsymbol{\mathcal{S}},\mathbf{L}} \ \|\overline{\boldsymbol{\mathcal{M}}}\|_* + \lambda\|\boldsymbol{\mathcal{E}}\|_1, \ s.t.\ \boldsymbol{\mathcal{Y}} = \overline{\boldsymbol{\mathcal{M}}} \times_3 \mathbf{L} + \boldsymbol{\mathcal{E}}, \mathbf{L}^T\mathbf{L} = \boldsymbol{I}, \\
&(\text{TC}): \max_{\overline{\boldsymbol{\mathcal{M}}},\mathbf{L}} \ \|\overline{\boldsymbol{\mathcal{M}}}\|_*, \ s.t.\ \boldsymbol{\mathcal{P}}_{\boldsymbol{\Omega}}(\boldsymbol{\mathcal{Y}}) = \boldsymbol{\mathcal{P}}_{\boldsymbol{\Omega}}(\overline{\boldsymbol{\mathcal{M}}} \times_3 \mathbf{L}), \mathbf{L}^T\mathbf{L} = \boldsymbol{I}.
\end{aligned} \tag{13}$$

3.2 Incoherence Conditions

The incoherence condition is one of the most vital theoretical tools in low-rank recovery [33, 3, 4]. Below, we define $\mathring{\mathfrak{e}}_i$ as the tensor column basis and the tensor incoherence conditions similar to [3].

Definition 7 (Tensor Incoherence Conditions) *For* $\boldsymbol{\mathcal{X}}_0 \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ *with t-SVD rank* R, *it has the skinny t-SVD* $\boldsymbol{\mathcal{X}}_0 = \boldsymbol{\mathcal{U}} *_L \boldsymbol{\mathcal{S}} *_L \boldsymbol{\mathcal{V}}^T$. *Then* $\boldsymbol{\mathcal{X}}_0$ *is said to satisfy the tensor incoherence conditions with parameter* μ *if*

$$\max_{i \in [1,n_1]} \|\boldsymbol{\mathcal{U}}^T *_L \mathring{\mathfrak{e}}_i\|_F \leq \sqrt{\frac{\mu R}{n_1}}, \max_{j \in [1,n_2]} \|\boldsymbol{\mathcal{V}}^T *_L \mathring{\mathfrak{e}}_j\|_F \leq \sqrt{\frac{\mu R}{n_2}}, \|\boldsymbol{\mathcal{U}} *_L \boldsymbol{\mathcal{V}}^T\|_F \leq \sqrt{\frac{\mu R}{n_1 n_2}}. \tag{14}$$

Algorithm 1 ADMM for solving ATNN-RPCA model (13)

Input: Observation $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, $\lambda = 1/\sqrt{\max(n_1, n_2)}$,$\mu = 1/\|\boldsymbol{\mathcal{Y}}\|_*$, $\rho = 1.25$, $\mu_m = 1e^7\mu$, and the column number of learnable COM matrix r_3.

Initialize $\boldsymbol{\Lambda} = \boldsymbol{\mathcal{E}} = \mathcal{O}$, $\overline{\boldsymbol{\mathcal{M}}} = \text{bdiag}(\boldsymbol{\mathcal{U}})$ and $\mathbf{L} = \mathbf{V}$, where $\boldsymbol{\mathcal{U}}, \mathbf{V}$ is the low-rank tensor decomposition of among mode-3, i.e., $\text{unfold}_3(\boldsymbol{\mathcal{Y}}) = (\boldsymbol{\mathcal{U}}) \times_3 \mathbf{L}$

while not convergence **do**

Update $\overline{\boldsymbol{\mathcal{M}}} := \text{SVD}_{1/\mu}((\boldsymbol{\mathcal{Y}} - \boldsymbol{\mathcal{E}} + \boldsymbol{\Lambda}/\mu) \times_3 \mathbf{L}^T)$.

Update $\mathbf{L} := \mathbf{B}\mathbf{D}^T$, where $[\mathbf{B}, \mathbf{C}, \mathbf{D}] = \text{svd}(\text{unfold}_3(\boldsymbol{\mathcal{Y}} - \boldsymbol{\mathcal{E}} + \boldsymbol{\Lambda}/\mu)^T \text{unfold}_3(\text{bfold}(\overline{\mathcal{U}})))$.

Update $\boldsymbol{\mathcal{X}} := \overline{\boldsymbol{\mathcal{M}}} \times_3 \mathbf{L}$

Update $\boldsymbol{\mathcal{E}} := \mathcal{S}_{\lambda/\mu}(\boldsymbol{\mathcal{Y}} - \boldsymbol{\mathcal{X}} + \boldsymbol{\Lambda}/\mu)$.

Update multipliers $\boldsymbol{\Lambda} := \boldsymbol{\Lambda} + \mu(\boldsymbol{\mathcal{Y}} - \boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{E}})$;

Let $\mu = \min\{\rho\mu, \mu_m\}$.

end while

Output: recovered tensors $\boldsymbol{\mathcal{X}} = \overline{\boldsymbol{\mathcal{M}}} \times_3 \mathbf{L}$ and $\boldsymbol{\mathcal{E}}$.

3.3 Main results

We now demonstrate that both the model (12) and (13) possess exact recovery capability.

Theorem 2 (TRPCA Theorem) *Consider ATNN-based TRPCA model (12) and (13). Suppose that $\boldsymbol{\mathcal{X}}_0 \in \mathbb{R}^{n \times n \times n_3}$ obeys the tensor incoherence conditions (14) and $\boldsymbol{\mathcal{E}}_0$'s support set, denoted as $\boldsymbol{\Omega}_0$, is uniformly distributed among all sets of cardinality m. Then, there exist universal constants $c_1, c_2 > 0$ such that $(\boldsymbol{\mathcal{X}}_0, \boldsymbol{\mathcal{E}}_0)$ is the unique solution to model (12) and (13) when $\lambda = 1/\sqrt{n}$ with probability at least $1 - c_1(nn_3)^{-c_2}$, provided that*

$$rank_t(\boldsymbol{\mathcal{X}}_0) \leq \rho_r \mu^{-1} n \log^{-2}(n) \ \ and \ \ m \leq \rho_s n^2 n_3, \tag{15}$$

where $\rho_r, \rho_s > 0$ are some numerical constants.

Theorem 3 (TC Theorem) *Consider ATNN-based TC model (12) and (13). Suppose that $\boldsymbol{\mathcal{X}}_0 \in \mathbb{R}^{n \times n \times n_3}$ obeys the tensor incoherence conditions (14) and $\boldsymbol{\Omega} \sim Ber(p)$. Then, there exist universal constants $c_0, c_1, c_2 > 0$ such that $\boldsymbol{\mathcal{X}}_0$ is the unique solution to model model (12) and (13) with probability at least $1 - c_1(nn_3)^{-c_2}$, provided that*

$$p \geq c_0 \mu R n^{-1} \log^2(n). \tag{16}$$

Remark 2 *It should be noted that although the model (12) and (13) are slightly different, they are the same in the proof of the exact recoverable theory. Assume that the optimal values of models (12) and (13) are $(\hat{\boldsymbol{\mathcal{X}}}, \hat{\boldsymbol{\mathcal{E}}})$ and $(\hat{\overline{\boldsymbol{\mathcal{M}}}}, \hat{\mathbf{L}}, \hat{\boldsymbol{\mathcal{E}}})$, respectively. A recoverable theory of model (12) requires proving $(\hat{\boldsymbol{\mathcal{X}}}, \hat{\boldsymbol{\mathcal{E}}}) = (\boldsymbol{\mathcal{X}}_0, \boldsymbol{\mathcal{E}}_0)$ under the given $\mathbf{L}$ in advance. A recoverable theory of model (13) requires proving $(\hat{\overline{\boldsymbol{\mathcal{M}}}} \times_3 \hat{\mathbf{L}}, \hat{\boldsymbol{\mathcal{E}}}) = (\boldsymbol{\mathcal{X}}_0, \boldsymbol{\mathcal{E}}_0)$ under the final learned $\hat{\mathbf{L}}$.*

3.4 Solving Algorithm

This subsection derives efficient algorithms for solving the ATNN-based TRPCA and TC problem via the Alternating Direction Method of Multipliers (ADMM) framework [34].

We first write the augmented Lagrangian function of the TRPCA problem in Eq. (13) as:

$$\min_{\overline{\boldsymbol{\mathcal{M}}}, \boldsymbol{\mathcal{E}}, \boldsymbol{\Lambda}, \mathbf{L}^T\mathbf{L} = \boldsymbol{I}} \|\overline{\boldsymbol{\mathcal{M}}}\|_* + \lambda \|\boldsymbol{\mathcal{E}}\|_1 + \frac{\mu}{2} \|\boldsymbol{\mathcal{Y}} - \overline{\boldsymbol{\mathcal{M}}} \times_3 \mathbf{L} - \boldsymbol{\mathcal{E}} + \boldsymbol{\Lambda}/\mu\|_F^2, \tag{17}$$

where μ is the penalty parameter and $\boldsymbol{\Lambda}$ is the lagrange multiplier.

Due to page limitation, we provide Algorithm 1 for solving Eq. (17) using the soft-thresholding operator $\mathcal{S}\tau(\cdot)$ [35] and the singular value soft-thresholding operator $\text{SVD}\tau(\cdot)$ [36]. Additionally, for the ATNN-TC model (13), we provide Algorithm 2 directly. For more detailed information, please refer to the supplementary material.

Algorithm 2 ADMM for solving ATNN-TC model (13)

Input: Observation $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ with support set $\boldsymbol{\Omega}$, $\mu = 0.1$, $\rho = 1.05$, $\mu_m = 1e^7\mu$, and the column number of learnable COM matrix r_3.

Similar initialization with Algorithm 1.

while not convergence **do**

Update $\overline{\boldsymbol{\mathcal{M}}}, \mathbf{L}, \boldsymbol{\mathcal{X}}, \boldsymbol{\Lambda}$ via the similar way in Algorithm 1.

Update $\boldsymbol{\mathcal{E}} := \boldsymbol{\mathcal{P}}_{\Omega}(\boldsymbol{\mathcal{Y}} - \boldsymbol{\mathcal{X}} + \boldsymbol{\Lambda}/\mu)$, where $\boldsymbol{\mathcal{P}}_{\boldsymbol{\Omega}}$ is projection operator.

Let $\mu = \min\{\rho\mu, \mu_m\}$.

end while

Output: recovered tensors $\boldsymbol{\mathcal{X}} = \overline{\boldsymbol{\mathcal{M}}} \times_3 \mathbf{L}$.

3.5 Computational Complexity Analysis

As depicted in Algorithm 1 and 2, each iteration of the algorithm involves updating $\overline{\boldsymbol{\mathcal{M}}}$ through small-scale SVD computations, updating $\mathbf{L}$ through small-scale SVD computation, updating $\boldsymbol{\mathcal{E}}$ through soft thresholding operations, and some matrix multiplications. For a third-order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the time complexity of the soft threshold operator is $\mathcal{O}(n_1 n_2 n_3)$, the time complexity of solving $\mathbf{L}$ is $\mathcal{O}(n_3 r_3^2)$, and the time complexity of solving $\overline{\boldsymbol{\mathcal{M}}}$ is $\mathcal{O}(r_3 n_1 n_2^2)$. Thus, the overall time complexity of Algorithm 1 and 2 is $\mathcal{O}(r_3 n_1 n_2^2 + n_3 r_3^2 + n_1 n_2 n_3)$. Similarly, for the DFT-transformed TRPCA and TC models, the time complexity is $\mathcal{O}(n_3 n_1 n_2^2 + n_1 n_2 n_3)$. By comparing the two time complexities mentioned above, it can be observed that their ratio is positively correlated with r_3/n_3. Therefore, as the low-rank property of the tensor in the third dimension becomes stronger, the acceleration capability of the proposed algorithm in this paper also becomes stronger.

4 Experiments

In this section, we present numerical experiments to validate the main results stated in Theorems 2 and 3. Following the suggestion of Theorem 2, we set $\lambda = 1/\sqrt{\max\{n_1, n_2\}}$ for the TRPCA task in all experiments. However, it should be noted that further performance improvements can be achieved by carefully tuning the value of λ. The suggested value in the theory provides a useful guideline in practical applications. All simulations were conducted on a PC equipped with an Intel(R) Core(TM) i5-10600KF 4.10GHz CPU, 32 GB memory, and a GeForce RTX 3080 GPU with 10 GB memory.

4.1 Simulated Experiments

In this section, we will verify the correct recovery guarantee of Theorem 2 and 3 on randomly generated problems. We generate a tensor with tubal rank R as a product $\boldsymbol{\mathcal{X}}_0 = \boldsymbol{\mathcal{P}} *_L \boldsymbol{\mathcal{Q}}^T$, where $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{Q}}$ are $n \times R \times n$ tensors with entries independently sampled from $\mathcal{N}(0, 1/n)$ distribution and the COM $\mathbf{L} \in \mathbb{R}^{r_3 \times n}$ is generated by orthogonalizing the random matrix with entries independently sampled from $\mathcal{N}(0, 1)$. For the TRPCA task, the support set $\boldsymbol{\Omega}$ (with size m) of $\boldsymbol{\mathcal{E}}_0$ with independent Bernoulli ± 1 entries is chosen uniformly at random, and the observation tensor is set as: $\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{X}}_0 + \boldsymbol{\mathcal{E}}_0$. For the TC tasks, the observation $\boldsymbol{\mathcal{Y}}$ is set as $\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{P}}_{\boldsymbol{\Omega}}(\boldsymbol{\mathcal{X}}_0)$.

Next, we investigate how the tubal rank of $\boldsymbol{\mathcal{X}}_0$ and the sparsity of $\boldsymbol{\mathcal{E}}_0$ (and missing ratio of $\boldsymbol{\mathcal{X}}_0$) affect the performance of model (12) and (13). We consider $n = 50$ and two values of r_3, i.e., $r_3 = 5, 20$. We vary the sparsity $\rho_s$of $\boldsymbol{\mathcal{E}}_0$ as $[0.01 : 0.01 : 0.5]$, the missing ratio ρ of $\boldsymbol{\mathcal{X}}_0$ as $[0.01 : 0.02 : 0.99]$, and tubal rank of $\boldsymbol{\mathcal{X}}_0$ as $[1 : 1 : 50]$, respectively. For each combination of (R, ρ_s) and (R, ρ), we perform 10 test instances and declare a trial successful if the recovered tensor $\hat{\boldsymbol{\mathcal{X}}}$ satisfies $\|\hat{\boldsymbol{\mathcal{X}}} - \boldsymbol{\mathcal{X}}_0\|_F / |\boldsymbol{\mathcal{X}}_0\|_F \leq 0.01$. The fraction of successful recoveries are plotted in Figure 1. From Figure 1, we observe that there is a significant region where the recovery is correct for both models. Furthermore, two notable phenomena can be observed from the figure:

1) The phase transition diagram in the first row of Figure 1 closely resembles the second row, indicating that even if we don't know the correct COM $\mathbf{L}$ in the model (12), we can learn the COM $\mathbf{L}$ through model (13).

2) The phase transition diagram of $r_3 = 5$ is much better than that of $r_3 = 20$ for both TRPCA and TC tasks, which shows that it is necessary to consider the low-rank property of mode 3.

Table 2: Quantitative comparison of all RPCA-based competing methods under salt-and-pepper noise with the variance of **0.6**. The best and second results are highlighted in bold italics and underline.

Methods	WDC			PaviaU			Beans			Cloth		
	PSNR	SSIM	Times	PSNR	SSIM	Times	PSNR	SSIM	Times	PSNR	SSIM	Times
RPCA	32.08	0.5223	28.99	24.98	0.8264	6.59	17.88	0.5920	**17.92**	18.47	0.5418	**18.28**
SNN	26.02	0.7178	136.2	31.34	0.9492	121.1	16.14	0.5238	176.2	16.77	0.5297	176.7
KBR	22.64	0.6438	167.2	20.91	0.4477	58.63	20.26	0.4162	252.1	20.91	0.5454	162.9
TNN	19.619	0.3728	419.2	17.09	0.2345	120.2	20.39	0.2572	322.4	15.51	0.1744	324.8
CTNN	17.21	0.2036	485.7	15.38	0.1163	130.7	15.64	0.1218	363.4	14.55	0.1162	353.9
CTV	33.85	0.9454	170.2	31.91	0.8872	41.85	29.35	0.7770	103.8	27.33	0.7721	102.2
TCTV	32.12	0.9090	815.2	29.62	0.8554	172.5	**32.85**	**0.9204**	641.3	27.36	0.7534	627.9
Ours	**39.82**	**0.9913**	**21.34**	**35.31**	**0.9721**	**5.32**	29.46	0.9108	29.22	**27.53**	**0.8563**	19.30

Table 3: Quantitative comparison of all competing methods under missing ratio with **0.95**. The best and second results are highlighted in bold italics and underline, respectively.

Methods	WDC			PaviaU			Beans			Cloth		
	PSNR	SSIM	Times	PSNR	SSIM	Times	PSNR	SSIM	Times	PSNR	SSIM	Times
LRMC	18.53	0.4623	**24.38**	15.17	0.2834	**2.93**	15.96	0.3972	**7.61**	13.11	0.1902	**10.95**
HaLRTC	22.09	0.6676	54.37	18.87	0.3912	30.34	20.62	0.4542	64.48	19.01	0.3570	92.65
KBR	31.42	0.9022	1589	29.92	0.8591	725.7	26.06	0.7208	1253	24.14	0.6422	1292
TNN	30.01	0.8824	1019	26.43	0.7126	207.9	26.10	0.6712	419.2	23.46	0.6012	441.2
CTNN	33.36	0.9432	378.9	31.69	0.9172	114.4	27.61	0.8041	129.6	25.71	0.7362	136.2
UTNN	27.89	0.8652	487.6	21.80	0.5982	156.3	17.28	0.4131	116.6	16.27	0.3183	117.9
FTNN	34.87	0.5320	4376	32.56	0.9092	1263	28.48	0.8143	1587	25.25	0.7253	2054
OITNN	32.92	0.9396	838.2	28.46	0.8142	292.4	27.28	0.7442	448.6	24.06	0.6516	391.8
TCTV	33.33	0.9391	2116	31.81	0.8960	861.4	**31.77**	**0.9143**	1570	28.38	0.8442	1488
S2NTNN	37.36	0.9749	168.7	**35.15**	**0.9431**	40.78	27.44	0.7589	104.2	**31.28**	**0.8679**	113.2
Ours	**38.06**	**0.9793**	232.4	33.94	0.9293	58.34	28.83	0.8164	156.3	25.81	0.7146	142.4

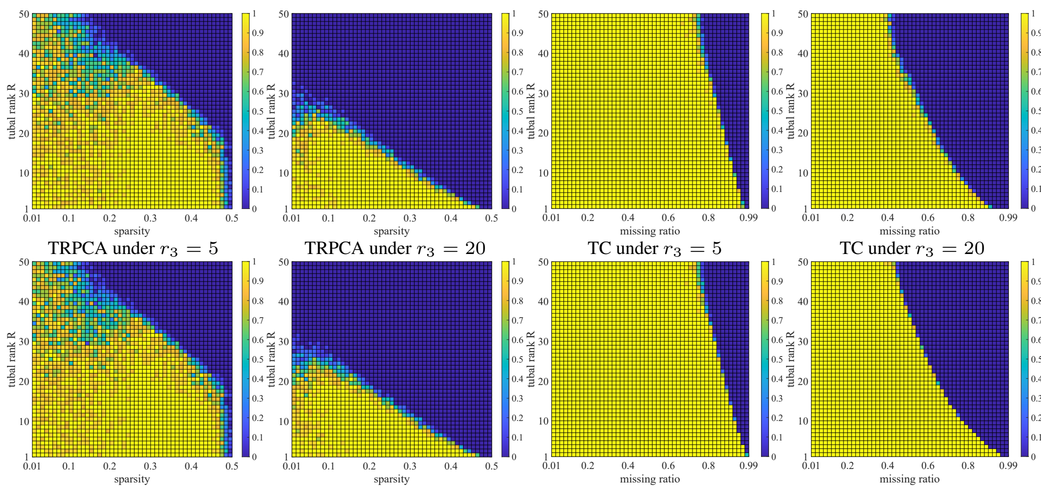


Figure 1: TRPCA and TC phase transition diagrams for varying tubal ranks of $\boldsymbol{\mathcal{X}}_0$ and sparsities of $\boldsymbol{\mathcal{E}}_0$ or missing ratio of $\boldsymbol{\mathcal{X}}_0$. The first and second rows show the phase transition diagrams based on models (13) and (12), respectively, under different r_3 settings.

4.2 Real Experiments

To validate the effectiveness of the proposed ATNN model in tensor recovery task, we conducted experiments on various datasets, including hyperspectral images (HSI), multispectral images (MSI), color video images, and surveillance videos. Due to page limitations, we have included the results of robustness analysis, parameter settings for robustness, convergence verification, and more detailed experimental outcomes in the Supplementary Material.

For comprehensive comparison, we have included additional state-of-the-art methods except those listed in Table 1. These methods include CTV [42] and TCTV [4] for the TRPCA task, LRMC [33], HaLRTC [1], UTNN [29], and OITNN [43] for the TC task, and GODEC [37], DECOLOR [38], OMoGMF [39], RegL1 [40], and PRMF [41] for background modeling. Before conducting this experiment, the gray value of each band was normalized into [0, 1] via the max-min formula.

Table 4: AUC comparison of all competing methods on all video sequences in the Li dataset. The best and second results in each video sequence are highlighted in bold italics and underline, respectively.

Methods	data										Time /s
	airp.	boot.	shop.	lobb.	esca.	curt.	camp.	wate.	foun.	Average	
RPCA [33]	0.8721	0.9168	0.9445	0.9130	0.9050	0.8722	0.8917	0.8345	0.9418	0.8991	2.37
GODEC [37]	0.9001	0.9046	0.9187	0.8556	0.9125	0.9131	0.8693	0.9370	0.9099	0.9023	0.64
DECOLOR [38]	0.8627	0.8910	0.9462	0.9241	0.9077	0.8864	**0.8945**	0.8000	0.9443	0.8952	8.29
OMoGMF [39]	0.9143	0.9238	0.9478	0.9252	0.9112	0.9049	0.8877	0.8958	0.9419	0.9170	3.92
RegL1 [40]	0.8977	**0.9249**	0.9423	0.8819	0.4159	0.8899	0.8871	0.8920	0.9194	0.8501	10.74
PRMF [41]	0.8905	0.9218	0.9415	0.8818	0.9065	0.8806	0.8865	0.8799	0.9166	0.9006	13.68
CTV [42]	0.9178	0.9107	**0.9541**	0.9337	0.9148	0.8710	0.8814	**0.9386**	0.9383	0.9180	10.28
TNN [2]	0.5218	0.5694	0.6605	0.6311	0.5981	0.5823	0.5464	0.6642	0.5781	0.5947	16.87
CTNN [28]	0.6859	0.6176	0.6835	0.6613	0.6582	0.6988	0.5881	0.5272	0.5450	0.6295	17.39
ATNN	**0.9185**	0.9227	0.9484	**0.9362**	**0.9158**	**0.9162**	0.8912	0.9152	**0.9456**	**0.9233**	**2.32**

Table 5: Quantitative comparison of all competing methods on color video under missing ratio with 0.95. The best and second results are highlighted in bold italics and underline, respectively.

Methods	Akiyo			Foreman			Carphone			News		
	PSNR	SSIM	Times	PSNR	SSIM	Times	PSNR	SSIM	Times	PSNR	SSIM	Times
LRMC	10.81	0.2626	**8.06**	8.79	0.1192	**7.21**	11.57	0.2713	**6.92**	13.27	0.3660	**13.41**
HaLRTC	17.66	0.5327	61.04	15.55	0.3336	44.87	14.20	0.3448	42.46	16.43	0.4890	87.63
KBR	29.76	0.9118	689.2	23.97	0.7193	668.2	26.49	0.8164	798.2	26.42	0.8480	1043
TNN	31.94	0.9343	217.5	23.15	0.6052	181.5	26.27	0.7658	493.6	28.56	0.8660	249.6
CTNN	28.63	0.8463	192.0	22.13	0.5779	152.7	25.06	0.7263	196.2	25.59	0.7740	174.7
UTNN	21.72	0.7237	172.4	16.51	0.2587	167.6	20.24	0.5394	202.7	21.21	0.7060	162.6
FTNN	30.74	0.9252	1258	22.97	0.6781	1123	25.43	0.7778	1335	28.77	0.8770	1494
OITNN	32.68	0.9533	397.5	23.89	0.7206	296.7	27.14	0.8340	472.3	29.43	0.9010	322.3
TCTV	33.41	0.9542	874.8	**26.69**	**0.8071**	821.4	**29.10**	**0.8747**	1103	**30.65**	**0.9170**	772.2
S2NTNN	33.16	0.9520	168.7	23.57	0.6091	83.98	27.33	0.8093	100.7	29.11	0.8872	90.61
Ours	**33.74**	**0.9574**	95.89	24.16	0.6252	78.21	27.44	0.7773	80.11	29.72	0.9021	78.94

4.2.1 Hyperspectral and Multispectral Image Recovery

Two HSI images, i.e., WDC [1] and PaviaU [2] datasets are used. The sizes of the two data are $256 \times 256 \times 191$ and $256 \times 256 \times 93$, respectively. Two MSI images in CAVE dataset [3], i.e., Cloth and Beans are used. The size of the two data is $512 \times 512 \times 31$.

For the TRPCA task, we conducted experiments with six different levels of salt and pepper noise variance: 0.1, 0.2, 0.3, 0.4, 0.5, and 0.6. Table 2 reports the performance metrics of each method under a variance of 0.6, demonstrating that our ATNN outperforms all competing methods. Notably, our method achieves superior performance despite only utilizing the low-rank property of tensors, surpassing the performance of CTV and TCTV, which additionally exploit the local smoothness and low-rank property of images. Furthermore, our method exhibits comparable computational efficiency to RPCA, indicating that the introduction of the learnable COM matrix effectively reduces the time complexity of the model. To better visualize the comparison, we choose three bands of HSI to form a pseudo-color image to show four representative competing methods' visual restoration performance, as shown in Figure 2. From the images, it is evident that our proposed ATNN model can effectively remove noise and preserve more detailed information.

For the TC task, since all the methods achieve very accurate recovery results when the sample ratio (SR) is high, we test four different SRs: 0.01, 0.05, 0.1 and 0.2. The metric of each tested algorithm under an SR of 0.05 is placed in Table 3. As can be seen from the metrics in the table, our proposed method excels in recovery performance and running time.

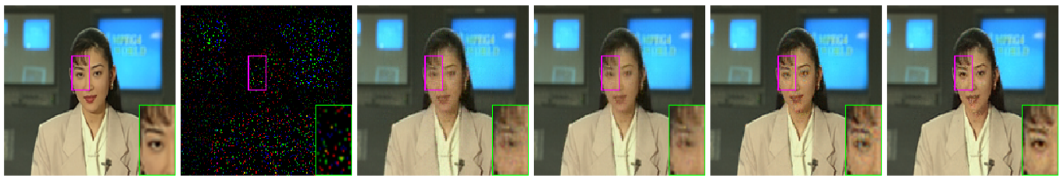
4.2.2 Background Modeling from Surveillance Video

The aim of this task is to separate the background and foreground from Surveillance Video. We choose nine video sequences in Li dataset [4] with the known foreground of size $144 \times 176 \times 20$ for testing, as shown in Table 4. It can be seen from the table that our proposed model is far ahead in

[1] https://engineering.purdue.edu/~biehl/MultiSpec/
[2] https://www.ehu.eus/ccwintco/index.php/
[3] https://www.cs.columbia.edu/CAVE/databases/multispectral/
[4] http://perception.i2r.a-star.edu.sg/bkmodel/bkindex.html

Clean: PSNR/SSIM | Noisy: 6.43/0.021 | RPCA: 29.64/0.932 | TNN: 19.85/0.355 | CTV: 33.57/0.943 | ATNN: 37.47/0.985

Figure 2: Denoised images of all competing methods with bands 58-27-9 as R-G-B under sparse noise with missing percent is 0.6 on simulated WDC dataset.



Clean: PSNR/SSIM | Observed: 6.24/0.014 | TNN: 31.66/0.935 | OITNN: 32.60/0.958 | S2NTNN: 35.27/0.966 | ATNN: 35.52/0.971

Figure 3: Recovered images of all competing methods under sample ratio of 0.05 on the 10th frame of Akiyo data.

terms of evaluation metrics and running time. Even compared to the CTV model that simultaneously utilizes local smoothness and low-rank priors, our method outperforms it. It is worth noting that although tensor-based models have a higher performance ceiling than matrix-based models due to their ability to capture more complex structures, for TNN regularization, if the variation matrix is not well defined, the results can even be worse than matrix-based methods. This further highlights the necessity of learning the transform matrix.

4.2.3 Color Video Completion

We selected four color video sequences, namely Akiyo, Foreman, Carphone, and Mobile, from the open-source YUV video dataset[5]. To ensure efficient comparison, we considered the first 100 frames of each color video sequence. As the color video is represented as a fourth-order tensor in RGB format with dimensions $144 \times 176 \times 3 \times 100$, we reshaped it into a tensor of size $144 \times 176 \times 300$. We adopted similar sample ratio (SR) settings as mentioned in Subsection 4.2.1. The performance metrics of all competing methods are presented in Table 5. It is evident that our proposed model consistently ranks within the top three, outperforming TCTV even under the Akiyo dataset. In comparison to other TNN models with fixed transform matrices, our model exhibits superior performance and remarkable computational efficiency. Furthermore, we provided the recovered images of some competing methods in Figure 3 for better visual comparison. For the convenience of observation, we have enlarged a part of the picture and placed the repair indicator below the picture. It can be seen that our proposed ATNN model has a strong ability to preserve the local information of the data.

5 Conclusion

In this paper, we introduce an efficient and learnable transformed tensor nuclear norm (TNN) model with a provable recovery guarantee. Our approach leverages the low-rank property of the third mode of the tensor to represent the tensor to be repaired as a combination of a small-sized tensor and a column-orthogonal matrix. The column-orthogonal matrix serves as an adaptively learned transform matrix derived from the data. By employing the nuclear norm on the small-sized tensor, our model achieves higher computational efficiency compared to existing methods. Additionally, we provide a theoretical framework that guarantees exact recovery for our proposed model with a column-orthogonal transform matrix. Extensive experimental results demonstrate the effectiveness of our approach and the validity of our theoretical findings.

Limitations There are two shortcomings in our work. Firstly, the recoverable theory does not explain how the low-rank property of the third dimension of the tensor affects the model's restoration performance. Secondly, the ATNN model only learns the low-rank property of the tensor, without incorporating image priors. These two points will be the focus of our future research.

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

References

[1] Ji Liu, Przemyslaw Musialski, Peter Wonka, and Jieping Ye. Tensor completion for estimating missing values in visual data. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):208–220, 2012.

[2] Zemin Zhang and Shuchin Aeron. Exact tensor completion using t-svd. *IEEE Transactions on Signal Processing*, 65(6):1511–1526, 2016.

[3] Canyi Lu, Jiashi Feng, Yudong Chen, Wei Liu, Zhouchen Lin, and Shuicheng Yan. Tensor robust principal component analysis with a new tensor nuclear norm. *IEEE transactions on pattern analysis and machine intelligence*, 42(4):925–938, 2019.

[4] Hailin Wang, Jiangjun Peng, Wenjin Qin, Jianjun Wang, and Deyu Meng. Guaranteed tensor recovery fused low-rankness and smoothness. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[5] Johann A Bengua, Ho N Phien, Hoang Duong Tuan, and Minh N Do. Efficient tensor completion for color image and video recovery: Low-rank tensor train. *IEEE Transactions on Image Processing*, 26(5):2466–2479, 2017.

[6] Wenrui Hu, Dacheng Tao, Wensheng Zhang, Yuan Xie, and Yehui Yang. The twist tensor nuclear norm for video completion. *IEEE transactions on neural networks and learning systems*, 28(12):2961–2973, 2016.

[7] Yao Wang, Jiangjun Peng, Qian Zhao, Yee Leung, Xile Zhao, and Deyu Meng. Hyperspectral image restoration via total variation regularized low-rank tensor decomposition. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(4):1227–1243, 2017.

[8] Hongyan Zhang, Lu Liu, Wei He, and Liangpei Zhang. Hyperspectral image denoising with total variation regularization and nonlocal low-rank tensor decomposition. *IEEE Transactions on Geoscience and Remote Sensing*, 58(5):3071–3084, 2019.

[9] Jiangjun Peng, Qi Xie, Qian Zhao, Yao Wang, Leung Yee, and Deyu Meng. Enhanced 3dtv regularization and its applications on hsi denoising and compressed sensing. *IEEE Transactions on Image Processing*, 29:7889–7903, 2020.

[10] Pan Zhou, Canyi Lu, Jiashi Feng, Zhouchen Lin, and Shuicheng Yan. Tensor low-rank representation for data recovery and clustering. *IEEE transactions on pattern analysis and machine intelligence*, 43(5):1718–1732, 2019.

[11] Jianlong Wu, Zhouchen Lin, and Hongbin Zha. Essential tensor learning for multi-view spectral clustering. *IEEE Transactions on Image Processing*, 28(12):5910–5922, 2019.

[12] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[13] Canyi Lu, Jiashi Feng, Zhouchen Lin, and Shuicheng Yan. Exact low tubal rank tensor recovery from gaussian measurements. *arXiv preprint arXiv:1806.02511*, 2018.

[14] Misha E Kilmer, Karen Braman, Ning Hao, and Randy C Hoover. Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging. *SIAM Journal on Matrix Analysis and Applications*, 34(1):148–172, 2013.

[15] Zemin Zhang, Gregory Ely, Shuchin Aeron, Ning Hao, and Misha Kilmer. Novel methods for multilinear data completion and de-noising based on tensor-svd. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3842–3849, 2014.

[16] Yang Mu, Ping Wang, Liangfu Lu, Xuyun Zhang, and Lianyong Qi. Weighted tensor nuclear norm minimization for tensor completion using tensor-svd. *Pattern Recognition Letters*, 130:4–11, 2020.

[17] Taixiang Jiang, Tingzhu Huang, Xile Zhao, and Liangjian Deng. Multi-dimensional imaging data recovery via minimizing the partial sum of tubal nuclear norm. *Journal of Computational and Applied Mathematics*, 372:112680, 2020.

[18] Hao Kong, Xingyu Xie, and Zhouchen Lin. t-schatten-p norm for low-rank tensor recovery. *IEEE Journal of Selected Topics in Signal Processing*, 12(6):1405–1419, 2018.

[19] Chunsheng Liu, Hong Shan, and Chunlei Chen. Tensor p-shrinkage nuclear norm for low-rank tensor completion. *Neurocomputing*, 387:255–267, 2020.

[20] Yang Zhou and YiuMing Cheung. Bayesian low-tubal-rank robust tensor factorization with multi-rank determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(1):62–76, 2019.

[21] Jian Lou and YiuMing Cheung. Robust low-rank tensor minimization via a new tensor spectral k-support norm. *IEEE Transactions on Image Processing*, 29:2314–2327, 2019.

[22] Hailin Wang, Feng Zhang, Jianjun Wang, Tingwen Huang, Jianwen Huang, and Xinling Liu. Generalized nonconvex approach for low-tubal-rank tensor recovery. *IEEE Transactions on Neural Networks and Learning Systems*, 33(8):3305–3319, 2021.

[23] Yisi Luo, Xile Zhao, Taixiang Jiang, Yi Chang, Michael K Ng, and Chao Li. Self-supervised nonlinear transform-based tensor nuclear norm for multi-dimensional image recovery. *IEEE Transactions on Image Processing*, 31:3793–3808, 2022.

[24] Hao Kong, Canyi Lu, and Zhouchen Lin. Tensor q-rank: new data dependent definition of tensor rank. *Machine Learning*, 110(7):1867–1900, 2021.

[25] Tongle Wu, Bin Gao, Jicong Fan, Jize Xue, and Wai Lok Woo. Low-rank tensor completion based on self-adaptive learnable transforms. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[26] Wenhao Xu, Xile Zhao, and Michael Ng. A fast algorithm for cosine transform based tensor singular value decomposition. *arXiv preprint arXiv:1902.03070*, 2019.

[27] Baburaj Madathil and Sudhish N George. Dct based weighted adaptive multi-linear data completion and denoising. *Neurocomputing*, 318:120–136, 2018.

[28] Canyi Lu, Xi Peng, and Yunchao Wei. Low-rank tensor completion with a new tensor nuclear norm induced by invertible linear transforms. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5996–6004, 2019.

[29] Michael K Ng, Xiongjun Zhang, and Xile Zhao. Patched-tube unitary transform for robust tensor completion. *Pattern Recognition*, 100:107181, 2020.

[30] Guangjing Song, Michael K Ng, and Xiongjun Zhang. Robust tensor completion using transformed tensor singular value decomposition. *Numerical Linear Algebra with Applications*, 27(3):e2299, 2020.

[31] Taixiang Jiang, Michael K Ng, Xile Zhao, and Tingzhu Huang. Framelet representation of tensor nuclear norm for third-order tensor completion. *IEEE Transactions on Image Processing*, 29:7233–7244, 2020.

[32] Jianli Wang, Tingzhu Huang, Xile Zhao, Taixiang Jiang, and Michael K Ng. Multi-dimensional visual data completion via low-rank tensor representation under coupled transform. *IEEE Transactions on Image Processing*, 30:3581–3596, 2021.

[33] Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):1–37, 2011.

[34] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

[35] David L Donoho. De-noising by soft-thresholding. *IEEE transactions on information theory*, 41(3):613–627, 1995.

[36] Jianfeng Cai, Emmanuel J Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on optimization*, 20(4):1956–1982, 2010.

[37] Tianyi Zhou and Dacheng Tao. Godec: Randomized low-rank & sparse matrix decomposition in noisy case. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*, 2011.

[38] Xiaowei Zhou, Can Yang, and Weichuan Yu. Moving object detection by detecting contiguous outliers in the low-rank representation. *IEEE transactions on pattern analysis and machine intelligence*, 35(3):597–610, 2012.

[39] Hongwei Yong, Deyu Meng, Wangmeng Zuo, and Lei Zhang. Robust online matrix factorization for dynamic background subtraction. *IEEE transactions on pattern analysis and machine intelligence*, 40(7):1726–1740, 2017.

[40] Yinqiang Zheng, Guangcan Liu, Shigeki Sugimoto, Shuicheng Yan, and Masatoshi Okutomi. Practical low-rank matrix approximation under robust l 1-norm. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1410–1417. IEEE, 2012.

[41] Naiyan Wang, Tiansheng Yao, Jingdong Wang, and Dit-Yan Yeung. A probabilistic approach to robust matrix factorization. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VII 12*, pages 126–139. Springer, 2012.

[42] Jiangjun Peng, Yao Wang, Hongying Zhang, Jianjun Wang, and Deyu Meng. Exact decomposition of joint low rankness and local smoothness plus sparse matrices. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[43] Andong Wang, QiBin Zhao, Zhong Jin, Chao Li, and GuoXu Zhou. Robust tensor decomposition via orientation invariant tubal nuclear norms. *Science China Technological Sciences*, 65(6):1300–1317, 2022.