Low Rank Tensor Completion Based on Self-Adaptive Learnable Transforms

Tongle Wu, Bin Gao, *Senior Member, IEEE*, Jicong Fan, Jize Xue, and W.L. Woo, *Senior Member, IEEE*

***Abstract*—The tensor nuclear norm (TNN), defined as the sum of nuclear norms of frontal slices of the tensor in a frequency domain, has been found useful in solving low rank tensor recovery problems. Existing TNN-based methods use either fixed or data-independent transformations, which may not be the optimal choices for the given tensors. As the consequence, these methods cannot exploit the potential low rank structure of tensor data adaptively. In this paper, we propose a framework called self-adaptive learnable transform (SALT) to learn a transformation matrix from the given tensor. Specifically, SALT aims to learn a lossless transformation that induces a lower average-rank tensor, where the Schatten-p quasi-norm is used as the rank proxy. Then because SALT is less sensitive to the orientation, we generalize SALT to other dimensions of tensor (SALTS), namely learning three self-adaptive transformation matrices simultaneously from given tensor. SALTS is able to adaptively exploit the potential low rank structures in all directions.We provide a unified optimization framework based on alternating direction multiplier method for SALTS model and theoretically prove the weak convergence property of proposed algorithm. Experimental results in hyperspectral image (HSI), color video, Magnetic Resonance imaging (MRI), and COIL-20 datasets show that SALTS is much more accurate in tensor completion than existing methods. Demo code can be found at https://faculty.uestc.edu.cn/gaobin/zh_CN/lwcg/153392/list/index.htm.**

***Index Terms*—self-adaptive, learnable transform, low rank, tensor completion**

I. Introduction

As a higher-order extension of matrix, N-th tensor ($N \geq 3$) is a useful mathematical form to represent a multiway array along N modes and has attracted considerable attention involving tasks of collection, processing, and analysis of high-dimensional data. It is essential to investigate robust and efficient algorithms to deal with tensors. The tensors often have low rank structures in many areas such as computer vision and neuroscience, which reveal that the data samples are generally embedded in low-dimensional manifolds [1]. A number of tensor methods utilizing the low rank structures have been successfully applied to practical problems [2], [3] such as color image and video processing [4]–[9], hyperspectral data recovery [10]–[13], collaborative filtering [14], recommender system design, [15] clustering, etc [16], [17].

There are various definitions of tensor rank originating from different tensor decomposition frameworks [18], which means that the rank of tensor is different from the rank of matrix. It is well known that the CP rank [19] and Tucker rank [20] based on the CANDECOMP/PARAFAC(CP) [21] decomposition and Tucker decomposition [22] respectively. These two kinds of tensor decompositions have been widely investigated and have achieved competitive performance in low rank tensor recovery. However, computing the CP rank is NP-hard [23], [24] and computing the Tucker rank requires unfolding tensors along each mode into matrices directly, which may destroy the intrinsic high-order interactive information.

An intuitive extension of matrix singular value decomposition (SVD) for tensor has been proposed for third-order tensor , which is known as tensor singular value decomposition (t-SVD) [25]. t-SVD was initially derived from the novel definition of tensor-tensor(t-t) product [26], which could operate integral third-order tensor rather than reshape the tensor into matrices [27]. Subsequently Kernfeld [28] discovered that t-t product based on circular convolution operator could be transformed into matrix-matrix product by Discrete Fourier Transform (DFT), which generates a transformed tensor in the frequency domain using DFT along the third mode. Based on the DFT transformed t-SVD, Zhang et al. [29], [30] proposed novel methods for multilinear data completion and denoising based on minimizing the tensor nuclear norm (TNN). Lu et al. [31] defined the tensor average rank and proved that a tensor always has low average rank if it has low tubal rank, which was defined in [28]. Lu et al. also revealed that is the convex envelope of the tensor average rank within the unit ball of the tensor spectral norm and proved the theoretical guarantee for the exact recovery in tensor robust principal component analysis (TRPCA). Following the DFT transformed t-SVD, a few variations of TNN such as weight TNN [32], partial sum of TNN [33], Schatten-p norm TNN [34], p-shrinkage TNN [35], tensor spectral k-support norm [36], have been proposed and tensor factorization methods based on t-t product derived from DFT have also been investigated in [17], [37]–[39].

In fact, three-dimensional data such as color images and videos always possess a notable ‘spatial-shifting’ correlation pattern and make similar spatial variation of data along temporal orientation smooth, which makes the DFT obtain low rankness in the frequency domain. [39], [40]. In this case, TNN based on fixed DFT transformation may be sensitive to data that violates the smooth pattern along the third mode. Fig.1 shows the PSNR of different videos varying with the

Tongle Wu is with the Chinese University of Hong Kong (Shenzhen) and Shenzhen Research Institute of Big Data Shenzhen, China. E-mail: tonglewu@std.uestc.edu.cn.

Bin Gao is with the University of Electronic Science and Technology of China. E-mail: bin_gao@uestc.edu.cn.

Jicong Fan is with the Chinese University of Hong Kong (Shenzhen) and Shenzhen Research Institute of Big Data Shenzhen, China. E-mail:fanjicong@cuhk.edu.cn.

Jize Xue is with the Northwestern Polytechnical University. E-mail: xuejize900507@mail.nwpu.edu.cn.

W.L. Woo is with the University of Northumbria at Newcastle. E-mail: wailok.woo@northumbria.ac.uk.

measurement of smoothness along frame direction. It can be observed that the less degree of smoothness the video has, the lower PSNR value tends to obtain. This phenomenon explains that the effect of completion for TNN is dependent on the smoothness of original data.

More specifically, for the data that is not smooth enough, the frequency domain tensor obtained by DFT does not satisfy fair low rank property, so the effect of completion based on low rank tensor methods will be degraded. The simple case is illustrated in **Lemma 1**, which shows a kind of tensors with low rank under identity transformation while with full rank under DFT transformation. In order to solve this problem, it is expected to replace the fixed DFT with other transforms that are not fixed but could be adaptively learned from the given data. Besides the fixed DFT transformations, another limitation of TNN is that the transformation is performed only along the third mode. Thus the orientation-sensitive transformation fails to capture the informative intra-mode and inter-mode correlations in other two modes. Obviously, the limitation becomes more significant for higher-order (e.g. $N \geq 4$) tensors. In this paper, our contributions are summarized as follows, which aims to overcome these limitations

- We propose to learn a adaptive transformation matrix from the given tensor data via minimizing the sum of the Schatten-p quasi-norms of the frontal slices of the transformed tensor. The learned transformation can induce a minimum-rank tensor in the transformation domain and hence has powerful potential in low rank tensor recovery.
- Due to the adaptive merit of learnable transformation, we generalize self-adaptive learnable transform (SALT) to the other modes of a given tensor to fully exploit the low rank, namely SALTS. SALTS overcomes the disadvantage of orientation sensitivity in TNN. More importantly, SALTS shows superior performance in tensor completion.
- We develop an efficient alternating direction multiplier method (ADMM) to solve the optimization of proposed model for tensor completion, and provide a weak convergence guarantee for the proposed algorithm, which is also validated by the numerical results. Results in synthetic data and several real datasets have further shown that the self-adaptive learnable transform based TNN achieves the state-of-the-art performance.

The rest of this paper is structured as follows. Section II introduces the notations, preliminaries , and related works. Section III presents the proposed method SALTS. Section IV consists of the experiments. Finally, we conclude the work in Section V.

II. NOTATIONS AND RELATED WORKS

A. Notations

In this section, we show the notations, definitions, and preliminaries which are necessary to present the proposed methods in this paper.

We use a, $\boldsymbol{a}$, $\boldsymbol{A}$, $\boldsymbol{\mathcal{A}}$ to denote scalar, vector, matrix , and tensor, respectively. The fields of real numbers and complex numbers are denoted as $\mathbb{R}$ and $\mathbb{C}$, respectively. Third-order tensors have column, row, and tube fibers, which are denoted by $\boldsymbol{\mathcal{A}}_{:ij}$, $\boldsymbol{\mathcal{A}}_{i:j}$, $\boldsymbol{\mathcal{A}}_{ij:}$, respectively. The i-th horizontal, lateral, and frontal slice of a third-order tensor, are presented as $\boldsymbol{\mathcal{A}}_{i::}$, $\boldsymbol{\mathcal{A}}_{:i:}$, $\boldsymbol{\mathcal{A}}_{::i}$, respectively. Specially, the i-th frontal slice can also be denoted as $\boldsymbol{A}^{(n)}$.

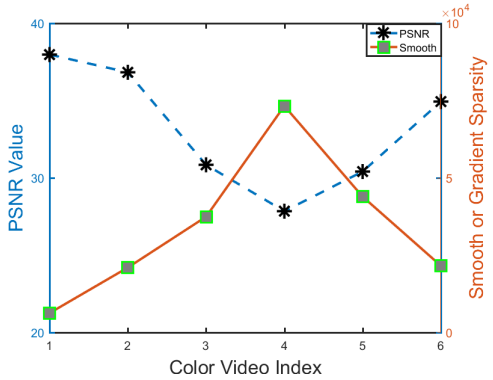


Fig. 1: The relationship between the effect (PSNR) of completion and smoothness along the dimension of video's frame with sampling rate 0.3. The test videos are listed in the order: 1: **Akiyo**; 2: **Bridge_close**; 3: **Carphone** ; 4: **Coastguard**; 5: **Suzie**; 6: **Miss-America** in YUV data. The smoothness is defined as: $\left\|\boldsymbol{L}\boldsymbol{X}_{(3)}\right\|_1$, $\boldsymbol{X}_{(3)} \in \mathbb{R}^{T\times hw}$, $L_{ii} = 1$ and $L_{ii+1} = -1$ for $i \in [1, T-1]$ other elements of $\boldsymbol{L}$ is 0.

The ℓ_1-norm and ℓ_∞ norm of a tensor $\boldsymbol{\mathcal{A}}^{n_1\times n_2\times n_3}$ is defined as $\left\|\boldsymbol{\mathcal{A}}\right\|_1 = \sum_{ijk}|\boldsymbol{\mathcal{A}}_{ijk}|$ and $\left\|\boldsymbol{\mathcal{A}}\right\|_\infty = \max_{ijk}|\boldsymbol{\mathcal{A}}_{ijk}|$. The inner product of two tensors is defined as $\langle\boldsymbol{\mathcal{A}},\boldsymbol{\mathcal{B}}\rangle = \sum_{ijk}\boldsymbol{\mathcal{A}}_{ijk}\boldsymbol{\mathcal{B}}_{ijk}$.The Frobenius norm of tensor is defined as $\left\|\boldsymbol{\mathcal{A}}\right\|_F = \sqrt{\sum_{ijk}\boldsymbol{\mathcal{A}}_{ijk}^2}$. We use $\left\|\boldsymbol{A}\right\|_* = \sum_i \sigma_i(\boldsymbol{A})$ to denote the nuclear norm of matrix $\boldsymbol{A}$, where $\sigma_i(\boldsymbol{A})$ is the i-th singular values of $\boldsymbol{A}$ in descent order. The Schatten-p (quasi) norm of matrix is defined as $\left\|\boldsymbol{A}\right\|_{S_p} = (\sum_i \sigma_i^p(\boldsymbol{A}))^{\frac{1}{p}}$, where $p > 0$. $\boldsymbol{a} = \text{diag}(\boldsymbol{A})$, which means $\boldsymbol{a}(i) = \boldsymbol{A}_{ii}$. $\boldsymbol{A} = \text{diag}(\boldsymbol{a})$, denotes that $\boldsymbol{A}_{ii} = \boldsymbol{a}(i)$ and other elements of $\boldsymbol{A}$ is 0.

The symbol $\otimes$ stands for the Kroneck product. The mode-n product of a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1\times I_2\cdots\times I_N}$ between a matrix $\boldsymbol{A} \in R^{J\times I_n}$ is represented as $\boldsymbol{\mathcal{B}} = \boldsymbol{\mathcal{X}} \times_n \boldsymbol{A}$ where $\boldsymbol{\mathcal{B}} \in \mathbb{R}^{I_1\times\cdots\times I_{n-1}\times J\times I_{n+1}\cdots\times I_N}$ and $[\boldsymbol{\mathcal{B}}]_{i_1\cdots i_{n-1}ji_{n+1}\cdots i_N} = \sum_{i_n=1}^{I_n} x_{i_1\cdots i_{n-1}i_n i_{n+1}\cdots i_N} a_{ji_n}$. The mode-$n$ matricization of $\boldsymbol{\mathcal{X}}$ is defined as $\boldsymbol{X}_{(n)} = unfold_n(\boldsymbol{\mathcal{X}})$, which is arranging the fibers $\boldsymbol{\mathcal{X}}_{i_1\cdots i_{n-1}:i_{n+1}\cdots i_N}$ as the columns of $\boldsymbol{X}_{(n)}$ and it's reverse operator is denoted by $\boldsymbol{\mathcal{X}} = Fold_n(Unfold_n(\boldsymbol{\mathcal{X}}))$. Then mode-$n$ product could be represented as $\boldsymbol{B}_{(n)} = \boldsymbol{A}\boldsymbol{X}_{(n)}$. The block circulant matrix $bcirc(\boldsymbol{\mathcal{A}}) \in \mathbb{R}^{n_1n_3\times n_2n_3}$ is defined as

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

Definition 1. ***(Trace norm (SNN))*** *[18] The trace norm of tensor is sum of nuclear norms (SNN), defined as:*

$$\left\|\boldsymbol{\mathcal{A}}\right\|_{SNN} = \sum_{n=1}^{N}\alpha_n\left\|\boldsymbol{A}_{(n)}\right\|_* \tag{2}$$

For $\boldsymbol{v} \in \mathbb{R}^n$, $\overline{\boldsymbol{v}} = DFT(\boldsymbol{v}) = \boldsymbol{F}_n\boldsymbol{v} \in \mathbb{C}^n$, where $\boldsymbol{F}_n$ represents an n-dimension DFT transform matrix. $[\boldsymbol{F_n}]_{jk} = (e^{-\frac{2\pi i}{n}})^{(j-1)*(k-1)}$ and $\boldsymbol{F_n^*F_n} = n\boldsymbol{I_n}$, $i = \sqrt{-1}$. fft($\boldsymbol{\mathcal{A}}$, [ ]) = $\overline{\boldsymbol{\mathcal{A}}} \in \mathbb{C}^{n_1\times n_2\times n_3}$ denotes performing the DFT on all the tubes of $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{n_1\times n_2\times n_3}$, whose mode-$n$ expression is $\overline{\boldsymbol{\mathcal{A}}} = \boldsymbol{\mathcal{A}} \times_3 \boldsymbol{F}_{n_3}$. The inverse fft on $\overline{\boldsymbol{\mathcal{A}}}$ can turn into original tensor, i.e. $\boldsymbol{\mathcal{A}}$=ifft($\overline{\boldsymbol{\mathcal{A}}}$, [ ], 3).

Definition 2. ***(Tensor Average Rank)*** *[31] For $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{n_1\times n_2\times n_3}$, the tensor average rank, denoted as* $\text{Rank}_a(\boldsymbol{\mathcal{A}})$ *is defined as:*

$$\text{Rank}_a(\boldsymbol{\mathcal{A}}) = \frac{1}{n_3}\sum_{k=1}^{n_3}\text{Rank}(\overline{\boldsymbol{A}}^{(n)}) \tag{3}$$

Definition 3. ***(Tensor Nuclear Norm)*** *[29] Tensor nuclear norm (TNN) of $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{n_1\times n_2\times n_3}$ is sum of nuclear norm of frontal slice matrices of DFT transformed tensor $\overline{\boldsymbol{\mathcal{A}}}$ defined as:*

$$\left\|\boldsymbol{\mathcal{A}}\right\|_{TNN} = \frac{1}{n_3}\sum_{n=1}^{n_3}\left\|\overline{\boldsymbol{A}}^{(n)}\right\|_* \tag{4}$$

Definition 4. ***(Transform Induced Tensor Average Rank)*** *[41] Let $\boldsymbol{L}$ be any invertible linear transform and satisfies $\boldsymbol{L}^T\boldsymbol{L} = \boldsymbol{L}\boldsymbol{L}^T = l\boldsymbol{I}_{n_3}$, tensor average rank of $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{n_1\times n_2\times n_3}$ induced by $\boldsymbol{L}$ is sum of rank of frontal slices of transformed tensor $\overline{\boldsymbol{\mathcal{A}}}_{\boldsymbol{L}} = \boldsymbol{\mathcal{A}} \times_3 \boldsymbol{L}$ defined as:*

$$\text{Rank}_t^L(\boldsymbol{\mathcal{A}}) = \frac{1}{n_3}\sum_{n=1}^{n_3}\text{Rank}(\overline{\boldsymbol{A}}_{\boldsymbol{L}}^{(n)}) \tag{5}$$

Definition 5. ***(Transform Induced Tensor Nuclear Norm)*** *[41] Let $\boldsymbol{L}$ be any invertible linear transform and satisfies $\boldsymbol{L}^T\boldsymbol{L} = \boldsymbol{L}\boldsymbol{L}^T = l\boldsymbol{I}_{n_3}$, tensor nuclear norm (TNN) of $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{n_1\times n_2\times n_3}$ induced by $\boldsymbol{L}$ is sum of nuclear norm of frontal sliced matrices of transformed tensor $\overline{\boldsymbol{\mathcal{A}}}_{\boldsymbol{L}} = \boldsymbol{\mathcal{A}} \times_3 \boldsymbol{L}$ defined as:*

$$\left\|\boldsymbol{\mathcal{A}}\right\|_{\boldsymbol{L}*} = \frac{1}{n_3}\sum_{n=1}^{n_3}\left\|\overline{\boldsymbol{A}}_{\boldsymbol{L}}^{(n)}\right\|_* \tag{6}$$

B. Related Works

Although there are many low rank methods for tensor completion, the intuition of proposed SALTS is derived from t-SVD decomposition. Thus, the introduction of related works mainly concentrates on t-SVD based low rank tensor completion.

Zhang et al. [29], [30] proposed novel methods for multi-linear data completion based on minimizing the tensor nuclear norm. Liu et.al [39] proposed fast Tubal-AltMin for the low-tubal-rank tensor completion problem and given the theoretical performance guarantee. Wang et al. [42] defined the generalized tensor Dantzig selector to recover a low-tubal-rank tensor from noisy linear measurements. Hou et al. [43] studied low-tubal-rank tensor recovery from binary measurements. Jiang [44] provided convex method for low-tubal-rank tensor completion with provable theoretical guarantee. Sun et al. [45] defined a novel generalized tensor tubal rank to tensor completion. Zhang et al. [46] given the minimal number of linear observations to reconstruct the low tubal rank tensor. Wang et al. [47] considered the structural difference between the observed data and missing data under t-SVD decomposition.

In fact, Kernfeld et al. [28] advocated that the t-t product can be modified to be equipped with any invertible transform which can replace the DFT transform with any invertible transform along the third mode. Following this research line, researchers in [48], [49] adopted Discrete Cosine Transform (DCT) which achieved superior performance than DFT in tensor completion. They proposed that any TNN induced from any invertible transform can be applied in TRPCA [50] and low rank tensor completion with exact recovery [41], but they did not give the specific transform. Authors in [51] proposed that unitary transform like Haar Wavelet transform could obtain sufficient low rank tensor in transformation domain. Recently, Jiang et al. [52] proposed framelet F-TNN that used the framelet representation of each tube to construct framelet transformed tensor. They advocated that the representation of each tube is represented sparsely due to redundancy of framelet basis. Then Jiang et al. [53] proposed low rank coefficient based on learnable dictionary derived from t-SVD framework. Although Kong et al. [54] have defined new tensor Q-rank based on novel data-dependent transformation for tensor completion, their data-dependent transformation can be learnt by principal component analysis, which is different with the proposed SALTS. In addition, they have not extended Q-rank to multi-modes and was only utilized in one dimension.

III. Proposed Methods

A. Motivation

In this section, we provide the following **Lemma 1** to show an example of the limitation when minimizing TNN of vanilla t-SVD with fixed Discrete Fourier Transform.

Lemma 1. *For the set $\boldsymbol{\mathcal{S}} = \{\text{Rank}(\boldsymbol{\mathcal{X}}(:,:,k)) = 1, \forall k \in [n] | \boldsymbol{\mathcal{X}} \in \mathbb{C}^{n\times n\times n}\}$, there always exists a tensor $\boldsymbol{\mathcal{X}} \in \boldsymbol{\mathcal{S}}$, which satisfies*

$$\text{Rank}_a(\boldsymbol{\mathcal{X}}) = \frac{1}{n}\sum_{k=1}^{n}\text{Rank}(\overline{\boldsymbol{X}}^{(n)}) = n \tag{7}$$

which means there always exists tensors $\in \mathbb{C}^{n\times n\times n}$, whose rank of each frontal slice is 1, while rank of each frontal slice of DFT transformed tensor is full n.

Proof. Our proof is constructive. Denote $\boldsymbol{A}$ as the random full rank matrix $\in \mathbb{R}^{n\times n}$. Then we construct a target tensor from frequency domain, $\overline{\boldsymbol{\mathcal{X}}}$ is constructed as follows:

$$\overline{\boldsymbol{X}}_{(3)} = \begin{bmatrix} \boldsymbol{A}(1,:) & \boldsymbol{A}(n,:) & \cdots & \boldsymbol{A}(2,:) \\ \boldsymbol{A}(2,:) & \boldsymbol{A}(1,:) & \cdots & \boldsymbol{A}(3,:) \\ \cdots & \cdots & \ddots & \cdots \\ \boldsymbol{A}(n,:) & \boldsymbol{A}(n-1,:) & \cdots & \boldsymbol{A}(1,:) \end{bmatrix} \tag{8}$$

It can be observed that vectorization of each frontal slice of $\overline{\boldsymbol{\mathcal{X}}}$ is corresponding to each row of $\boldsymbol{A}$, which equals cyclic queues of $\boldsymbol{A}$. So we could get $\text{Rank}_a(\boldsymbol{\mathcal{X}}) = \frac{1}{n}\sum_{k=1}^{n}\text{Rank}(\overline{\boldsymbol{X}}^{(n)}) = n$. Then we should verify that the tensor $\boldsymbol{\mathcal{X}}$ in the frequency

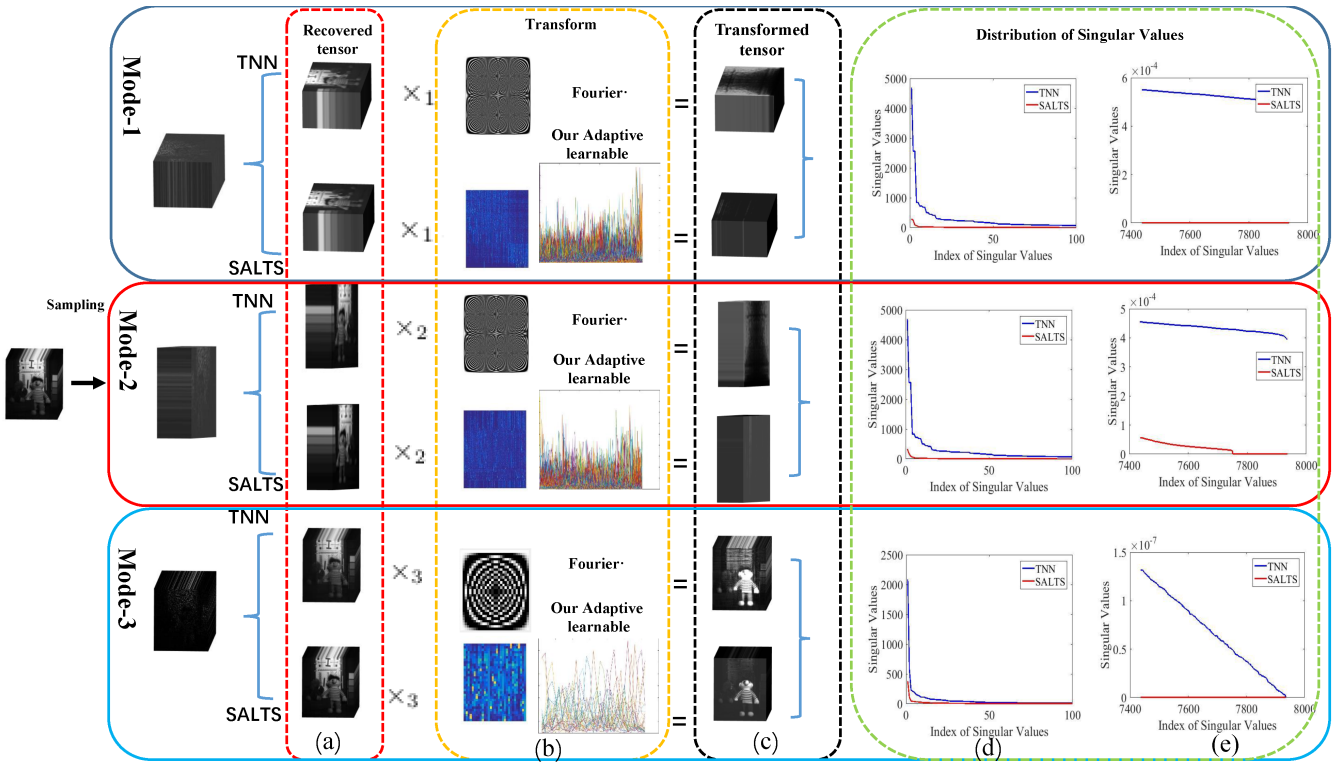


Fig. 2: Illustration of the framework of proposed SALTS compared with TNN. (a) Recovery tensor for three modes; (b) Fixed Fourier tansform and learned adaptive transforms from our algorithms. (c) Tensors in transformed domain. (d) The distribution of the top 100 largest singular values. (e) The distribution of the last 500 smallest singular values. Here, the singular values are collecting from each frontal slice of transformed tensors and are sorted in descending order.

domain constructed above as $\overline{\boldsymbol{\mathcal{X}}}$ is in the set $\boldsymbol{\mathcal{S}}$ by the inverse Fourier transform.

$$\begin{aligned} \boldsymbol{\mathcal{X}} &= ifft(\overline{\boldsymbol{\mathcal{X}}}, [], 3) = \overline{\boldsymbol{\mathcal{X}}} \times_3 \boldsymbol{F}^* \\ \boldsymbol{X}_{(3)} &= \boldsymbol{F}^* \overline{\boldsymbol{X}}_{(3)} \end{aligned} \tag{9}$$

Defining a new tensor $\boldsymbol{\mathcal{T}} \in \mathcal{R}^{1\times n\times n}$ whose frontal slice is

$$\boldsymbol{\mathcal{T}}(:,:,i) = \boldsymbol{A}(i,:) \tag{10}$$

Then we could get $\overline{\boldsymbol{X}}_{(3)} = bcirc(\boldsymbol{\mathcal{T}})$, further the block circulant matrix can be block diagonalize:

$$\begin{aligned} &\boldsymbol{F}_n bcirc(\boldsymbol{\mathcal{T}})(\boldsymbol{F}_n^{-1} \otimes \boldsymbol{I}_n) = \\ &\begin{bmatrix} \overline{\boldsymbol{T}}(:,:,1) & & & \\ & \overline{\boldsymbol{T}}(:,:,2) & & \\ & & \ddots & \\ & & & \overline{\boldsymbol{T}}(:,:,n) \end{bmatrix} \end{aligned} \tag{11}$$

we can conclude that there exists $\boldsymbol{\mathcal{X}}$

$$\text{Rank}(\boldsymbol{\mathcal{X}}(:,:,k)) = 1, then \quad \sum_{k=1}^{n} \text{Rank}(\boldsymbol{\mathcal{X}}(:,:,k)) = n, \tag{12}$$

□

Lemma 1 exposes the shortcoming of fixed Discrete Fourier Transform. There exists kinds of tensors which is full average rank in transformed domain by DFT, which prevents the tensor nuclear norm(TNN) based methods [29]–[31] from these tensor sets.

B. Self-Adaptive Learnable Transform along Mode-3 (SALT-3) for Tensor Completion

The novelty of this paper is to learn self-adaptive transform $\boldsymbol{L}$ and conduct the low rank tensor recovery simultaneously under a unified optimization framework, which is achieved by minimizing Rank_t^L induced by $\boldsymbol{L}$. Concretely, we learn $\boldsymbol{L}$ and low rank induced tensor as follows:

$$\begin{aligned} \min_{\boldsymbol{\mathcal{Y}},\boldsymbol{L},\boldsymbol{\mathcal{Z}}} \ & \text{Rank}_t^L(\boldsymbol{\mathcal{Y}}) + f(\boldsymbol{\mathcal{Y}}, \boldsymbol{\mathcal{Z}}) \\ \text{s.t. } & \boldsymbol{\mathcal{Y}} \in C_y, \ \boldsymbol{\mathcal{Z}} \in C_z, \end{aligned} \tag{13}$$

where C_y and C_z are constraint sets. In (13), we aim to learn a transform from the data itself to minimize the SALT induced rank of $\boldsymbol{\mathcal{Y}}$ plus a general function on $\boldsymbol{\mathcal{Y}}$ and $\boldsymbol{\mathcal{Z}}$. More specifically, $\boldsymbol{\mathcal{Y}}$ denotes the structured tensor which needs to be solved by (13) where $\boldsymbol{\mathcal{Z}}$ is the observed corrupted tensor data. Our intention is to illustrate that model (13) can be employed to obtain the recovered tensor $\boldsymbol{\mathcal{Y}}$ when given the corrupted tensor $\boldsymbol{\mathcal{Z}}$. The first term in (13) utilizes the low rank structure prior under adaptive transform L, while the second term guarantees that recovered $\boldsymbol{\mathcal{Y}}$ should fit the observed corrupted $\boldsymbol{\mathcal{Z}}$. Problem (13) covers a few important topics such as low rank tensor completion [30] and tensor robust PCA [31]. In this paper, we focus on the application of SALT in tensor completion. Given an incomplete $\boldsymbol{\mathcal{D}} \in \mathbb{R}^{n_1\times n_2\times n_3}$, we

propose to solve

$$\begin{aligned} &\min_{\boldsymbol{\mathcal{Y}},\boldsymbol{L}} \ \text{Rank}_t^L(\boldsymbol{\mathcal{Y}}) \\ &\text{s.t. } \pi_\Omega(\boldsymbol{\mathcal{Y}}) = \pi_\Omega(\boldsymbol{\mathcal{D}}), \end{aligned} \tag{14}$$

where $\pi_\Omega : \mathbb{R}^{n_1\times n_2\times n_3} \to \mathbb{R}^{n_1\times n_2\times n_3}$ is a linear operator that keeps the entries in Ω unchanged and sets those outside Ω(i.e., in Ω^c) to zero. According to (5), we could reformulate (14) into:

$$\begin{aligned} \min_{\boldsymbol{\mathcal{Y}},\boldsymbol{\mathcal{X}},\boldsymbol{L},} \ & \sum_{n=1}^{k} \text{Rank}(\boldsymbol{X}^{(n)}) \\ \text{s.t. } & \pi_\Omega(\boldsymbol{\mathcal{Y}}) = \pi_\Omega(\boldsymbol{\mathcal{D}}) \\ & \boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{Y}} \times_3 \boldsymbol{L}^T \\ & \left\|\boldsymbol{\mathcal{Y}}\right\|_F = \left\|\boldsymbol{\mathcal{X}}\right\|_F \end{aligned} \tag{15}$$

$\boldsymbol{L} \in R^{n_3\times k}$ satisfies $\boldsymbol{L}^T\boldsymbol{L} = \boldsymbol{I}_k$, where$k \le n_3$. It is worth noting that there are main three different points with (5). 1) In (5), the $\boldsymbol{L}$ is fixed, predefined and independent with tensor completion task. Our $\boldsymbol{L}$ is needed to learn by solving (15) to minimize the induced tensor average rank. Although [54] also proposed data-dependent, learnable transform $\boldsymbol{L}$, our learning approach is different. 2) In (5), $\boldsymbol{L}$ is square matrix and satisfies $\boldsymbol{L}^T\boldsymbol{L} = \boldsymbol{L}\boldsymbol{L}^T = l\boldsymbol{I}_{n_3}$, however, our $\boldsymbol{L}$ is not necessarily a square matrix, the k we set may $k < n_3$. We only require $\boldsymbol{L}$ as column orthogonal, which would reduce the computation of multiple SVD. 3) We add another constraint that $\left\|\boldsymbol{\mathcal{X}}\right\|_F = \left\|\boldsymbol{\mathcal{Y}}\right\|_F$, which keeps the power and information of original tensor from being lost after being transformed by $\boldsymbol{L}$. The last two equality constraints may not hold simultaneously in (15). The following lemma gives the condition that (15) has feasible solution, which guides the setting of k.

Lemma 2. *A necessary condition that* $\exists \boldsymbol{L}$ *makes* $\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{Y}} \times_3 \boldsymbol{L}^T$ *and* $\left\|\boldsymbol{\mathcal{Y}}\right\|_F = \left\|\boldsymbol{\mathcal{X}}\right\|_F$ *achieve simultaneously is* $k \ge \text{Rank}(\boldsymbol{Y}_{(3)})$.

In addition, $\boldsymbol{L}$ is easy to solve in the optimization, which will be verified later. Because of the presence of the rank function, it is NP-hard to solve (14) or (15). We use the Schatten-p quasi-norm as a surrogate function of rank and consider the following problem instead of (15):

$$\begin{aligned} \min_{\boldsymbol{\mathcal{Y}},\boldsymbol{\mathcal{X}},\boldsymbol{L},} \ & \sum_{n=1}^{k} \left\|\boldsymbol{X}^{(n)}\right\|_{S_p}^p \\ \text{s.t. } & \pi_\Omega(\boldsymbol{\mathcal{Y}}) = \pi_\Omega(\boldsymbol{\mathcal{D}}) \\ & \boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{Y}} \times_3 \boldsymbol{L}^T \\ & \left\|\boldsymbol{\mathcal{Y}}\right\|_F = \left\|\boldsymbol{\mathcal{X}}\right\|_F \end{aligned} \tag{16}$$

This is similar to the definition of t-Schatten-p norm in [34] while their t-Schatten-p norm is based on DFT framework which is different from our SALT paradigm. In addition, the proposed optimization approach to minimizing the SALT is different with theirs. We choose $0 < p \le 1$ in our model since this kind of non-convex norm can achieve a tighter bound in approximating the tensor multi-rank and obtain higher accuracy of solution [55], [56], which would guarantee the sufficient low rank of the transformed tensor.

C. From SALT-3 to SALTS

The disadvantage of t-SVD is sensitivity of the orientation due to the orientation of the low tubal rank definition. In real multi-way data like images and videos, there are ubiquitous "spatial-shifting" correlations (color channels and temporal frames) making such data spatially smooth along the third mode. This pattern can be exploited via performing DFT along the third mode to get the low-multi-rank transformed tensor [39] but the complex intra-mode and inter-mode correlations in the other two orientations have not been explored [40]. Although work in [57] proposed a method called sum of tensor nuclear norm (STNN) that applied DFT to all modes of a third-order tensor, the smooth pattern may not always exist or be captured by DFT in other two modes, which also makes STNN lack of physical explanation on real datasets.

Since the proposed SALT-3 is merited by adaptivity, the learned transformation is less orientation sensitive than DFT. We generalize the SALT-3 to the other two orientations with two main aims: overcoming the orientation sensitivity inherent with DFT and exploiting the intrinsic low rank property along the multi-orientations. This improves the limited representation ability and flexibility of TNN.

The relaxation model for learning the SALTS and achieving tensor completion extends from SALT-3 (16) to SALTS as follows:

$$\begin{aligned} \min_{\boldsymbol{\mathcal{Z}},\{\boldsymbol{\mathcal{Y}}_i\},\{\boldsymbol{L}_i\}} \ & \frac{\lambda}{2}\sum_{i=1}^{3}\left\|\boldsymbol{\mathcal{Z}} - \boldsymbol{\mathcal{Y}}_i\right\|_F^2 + \left\|\boldsymbol{\mathcal{Y}}_i \times_i \boldsymbol{L}_i^T\right\|_{s_p}^p \\ \text{s.t. } & \left\|\boldsymbol{\mathcal{Y}}_i \times_i \boldsymbol{L}_i^T\right\|_F = \left\|\boldsymbol{\mathcal{Y}}_i\right\|_F, \\ & \boldsymbol{L}_i^T\boldsymbol{L}_i = \boldsymbol{I}_i, \ \pi_\Omega(\boldsymbol{\mathcal{Z}}) = \pi_\Omega(\boldsymbol{\mathcal{D}}) \end{aligned} \tag{17}$$

where $\boldsymbol{\mathcal{D}} \in \mathbb{R}^{n_1\times n_2\times n_3}$ is observed incomplete tensor. $\boldsymbol{L}_i \in \mathbb{R}^{n_i\times k_i}$ satisfying $\boldsymbol{L}_i^T\boldsymbol{L}_i = \boldsymbol{I}_{k_i} (i = 1, 2, 3)$ are the SALTS learned by (17) along all modes. Model (16) only apply transformation to third mode, while (17) generalizes the (16) by applying three transformations to all modes of tensor simultaneously.

The integral framework of our proposed SALTS for tensor completion and illustration of it's superiority are presented in Fig.2. To understand the advantage of SALTS for tensor recovery, we show the distributions of singular values of all the frontal slices in the transformed tensor. As an example, we conduct experiment on the **chart and stuffed toy** in CAVE data with sample rate 0.1 in tensor completion task. The input tensor size is $256 \times 256 \times 31$ and pixels are normalized into (0,1). The Fig.2 illustrate that singular values in SALTS are much smaller than that of Fourier transform domain, which reveals that transformed tensor via SALTS would achieve and ensure sufficient and better low rank property.

In the next part, we will derive the optimization algorithm based on alternative direction multiplier method (ADMM) to solve the constrained minimization problems for relaxation of (16) and (17). Because the relaxation of (16) is a special case with only one SALT, so we only provide the optimization algorithm of (17).

D. Optimization Algorithm Based on ADMM

Actually, due to the orthogonal constraint on $\boldsymbol{L}_i$, it is a challenge to solve above constrained optimization problem. Usually, solving the $\boldsymbol{L}_i$ needs to apply manifold optimization method, which is time consuming. However, according to **theorem 1** in [58], the optimization problem of (17) is equivalent to the following optimization model, which means that the solutions of the two optimization problems are equivalent.

$$\begin{aligned}\min_{\{\boldsymbol{\mathcal{X}}_i,\boldsymbol{\mathcal{Y}}_i,\boldsymbol{L}_i\},\boldsymbol{\mathcal{Z}}} \ & \frac{\lambda}{2}\sum_{i=1}^{3}\left\|\boldsymbol{\mathcal{Z}}-\boldsymbol{\mathcal{Y}}_i\right\|_F^2+\sum_{i=1}^{3}\left\|\boldsymbol{\mathcal{X}}_i\right\|_{S_p}^p \\ \text{s.t.} \ & \boldsymbol{\mathcal{X}}_i\times_i\boldsymbol{L}_i=\boldsymbol{\mathcal{Y}}_i,\ \boldsymbol{L}_i^T\boldsymbol{L}_i=\boldsymbol{I}_i, \\ & \pi_\Omega(\boldsymbol{\mathcal{Z}})=\pi_\Omega(\boldsymbol{\mathcal{D}}).\end{aligned} \tag{18}$$

where $\boldsymbol{\mathcal{X}}_1=\boldsymbol{\mathcal{Y}}_1\times_1\boldsymbol{L}_1\in\mathbb{R}^{k_1\times n_2\times n_3}$, $\boldsymbol{\mathcal{X}}_2=\boldsymbol{\mathcal{Y}}_2\times_2\boldsymbol{L}_2\in\mathbb{R}^{n_1\times k_2\times n_3}$, $\boldsymbol{\mathcal{X}}_3=\boldsymbol{\mathcal{Y}}_3\times_3\boldsymbol{L}_3\in\mathbb{R}^{n_1\times n_2\times k_3}$ are transformed tensors, $\left\|\boldsymbol{\mathcal{X}}_1\right\|_{s_p}^p=\sum_{n=1}^{k_1}\left\|\boldsymbol{\mathcal{X}}_{n::}\right\|_{s_p}^p$, $\left\|\boldsymbol{\mathcal{X}}_2\right\|_{s_p}^p=\sum_{n=1}^{k_2}\left\|\boldsymbol{\mathcal{X}}_{:n:}\right\|_{s_p}^p$, $\left\|\boldsymbol{\mathcal{X}}_3\right\|_{s_p}^p=\sum_{n=1}^{k_3}\left\|\boldsymbol{\mathcal{X}}_{::n}\right\|_{s_p}^p$. The corresponding augmented Lagrange formulation converts the original constrained optimization problem into an unconstrained optimization problem as follows

$$\mathcal{L}_\mu(\boldsymbol{\mathcal{Z}},\{\boldsymbol{\mathcal{Y}}_i,\boldsymbol{L}_i,\boldsymbol{\mathcal{X}}_i\}_{i=1}^3)=\sum_{i=1}^{3}\langle\boldsymbol{\mathcal{P}}_i,\boldsymbol{\mathcal{X}}_i\times_i\boldsymbol{L}_i-\boldsymbol{\mathcal{Y}}_i\rangle \tag{19}$$

$$+\frac{\mu}{2}\left\|\boldsymbol{\mathcal{X}}_i\times_i\boldsymbol{L}_i-\boldsymbol{\mathcal{Y}}_i\right\|_F^2+\frac{\lambda}{2}\left\|\boldsymbol{\mathcal{Z}}-\boldsymbol{\mathcal{Y}}_i\right\|_F^2+\left\|\boldsymbol{\mathcal{X}}_i\right\|_{S_p}^p. \tag{20}$$

$\boldsymbol{\mathcal{P}}_i(i=1,2,3)$ are Lagrange multipliers and μ is an increased positive penalty in iterative steps to achieve the holding of equality constraints.

(1) **Update** $\{\boldsymbol{\mathcal{Y}}_i\}_{i=1}^3$

$$\boldsymbol{\mathcal{Y}}_i^{k+1}=\arg\min_{\boldsymbol{\mathcal{Y}}_i}\frac{\mu_k}{2}\left\|\boldsymbol{\mathcal{X}}_i^k\times_i\boldsymbol{L}_i^k-\boldsymbol{\mathcal{Y}}_i+\frac{\boldsymbol{\mathcal{P}}_i^k}{\mu_k}\right\|_F^2+\frac{\lambda}{2}\left\|\boldsymbol{\mathcal{Z}}^k-\boldsymbol{\mathcal{Y}}_i\right\|_F^2 \tag{21}$$

This quadratic objective function has an unique closed-form solution, set the derivative of (19) with respect to $\boldsymbol{\mathcal{Y}}_i^{k+1}$ and derive from first order optimal condition:

$$\boldsymbol{\mathcal{Y}}_i^{k+1}=\frac{\mu_k(\boldsymbol{\mathcal{X}}_i^k\times_i\boldsymbol{L}_i^k+\frac{\boldsymbol{\mathcal{P}}_i^k}{\mu_k})+\lambda\boldsymbol{\mathcal{Z}}^k}{\mu_k+\lambda} \tag{22}$$

(2) **Update** $\{\boldsymbol{L}_i\}_{i=1}^3$

$$\begin{aligned}\boldsymbol{L}_i^{k+1}&=\arg\min_{\boldsymbol{L}_i^T\boldsymbol{L}_i=\boldsymbol{I}_{k_i}}\frac{\mu}{2}\left\|\boldsymbol{\mathcal{X}}_i^k\times_i\boldsymbol{L}_i-\boldsymbol{\mathcal{Y}}_i^{k+1}+\frac{\boldsymbol{\mathcal{P}}_i^k}{\mu_k}\right\|_F^2 \\ &=\arg\max_{\boldsymbol{L}_i^T\boldsymbol{L}_i}\langle(\boldsymbol{Y}_{i(i)}^{k+1}-\frac{\boldsymbol{P}_{i(i)}^k}{\mu_k})\boldsymbol{X}_{i(i)}^{k^T},\boldsymbol{L}_i\rangle\end{aligned} \tag{23}$$

let $[\boldsymbol{U}_i^{k+1},\boldsymbol{S}_i^{k+1},\boldsymbol{V}_i^{k+1}]=SVD((\boldsymbol{Y}_{i(i)}^{k+1}-\frac{\boldsymbol{P}_{i(i)}^k}{\mu_k})\boldsymbol{X}_{i(i)}^{k^T})$. From the solution of well-known Orthogonal Procrustes [59]:

$$\boldsymbol{L}_i^{k+1}=\boldsymbol{U}_i^{k+1}\boldsymbol{V}_i^{k+1^T} \tag{24}$$

(3) **Update** $\{\boldsymbol{\mathcal{X}}_1,\boldsymbol{\mathcal{X}}_2,\boldsymbol{\mathcal{X}}_3\}$

$$\boldsymbol{\mathcal{X}}_i^{k+1}=\arg\min_{\boldsymbol{\mathcal{X}}_i}\left\|\boldsymbol{\mathcal{X}}_i\right\|_{sp}^p+\frac{\mu_k}{2}\left\|\boldsymbol{\mathcal{X}}_i\times_i\boldsymbol{L}_i^{k+1}-\boldsymbol{\mathcal{Y}}_i^{k+1}+\frac{\boldsymbol{\mathcal{P}}_i^k}{\mu_k}\right\|_F^2 \tag{25}$$

It is non-trial to use proximal operator because there is a linear transform $\boldsymbol{\mathcal{L}}$ in front of $\boldsymbol{\mathcal{X}}$

$$\begin{aligned}f(\boldsymbol{\mathcal{X}}_i)=&\frac{\mu_k}{2}\left\|\boldsymbol{\mathcal{X}}_i\times_i\boldsymbol{L}_i^{k+1}-\boldsymbol{\mathcal{Y}}_i^{k+1}+\frac{\boldsymbol{\mathcal{P}}_i^k}{\mu_k}\right\|_F^2\le f(\boldsymbol{\mathcal{X}}_i^k)+ \\ &\langle(\boldsymbol{\mathcal{X}}_i^k\times_i\boldsymbol{L}_i^k-\boldsymbol{\mathcal{Y}}_i^{k+1}+\frac{\boldsymbol{\mathcal{P}}_i^k}{\mu_k})\times_i\boldsymbol{L}_i^{k+1^T}\mu_k,\boldsymbol{\mathcal{X}}_i \\ &-\boldsymbol{\mathcal{X}}_i^k\rangle+\frac{\mu_k}{2}\left\|\boldsymbol{\mathcal{X}}_i-\boldsymbol{\mathcal{X}}_i^k\right\|^F \\ =&f(\boldsymbol{\mathcal{X}}_i^k)+\frac{\mu_k}{2}\left\|\boldsymbol{\mathcal{X}}_i-(\boldsymbol{\mathcal{Y}}_i^{k+1}-\frac{\boldsymbol{\mathcal{P}}_i^k}{\mu_k})\times_i\boldsymbol{L}_i^{k+1^T}\right\|^F\end{aligned} \tag{26}$$

Then, instead of (25), we use

$$\boldsymbol{\mathcal{X}}_i^{k+1}=\arg\min_{\boldsymbol{\mathcal{X}}_i}\left\|\boldsymbol{\mathcal{X}}_i\right\|_{Sp}^p+\frac{\mu_k}{2}\left\|\boldsymbol{\mathcal{X}}_i-\boldsymbol{\mathcal{A}}_i^k\right\|_F^2, \tag{27}$$

where $\boldsymbol{\mathcal{A}}_i^k=(\boldsymbol{\mathcal{Y}}_i^{k+1}-\frac{\boldsymbol{\mathcal{P}}_i^k}{\mu_k})\times_i\boldsymbol{L}_i^{k+1^T}$ and $i\in\{1,2,3\}$. Specifically, we have

$$\boldsymbol{\mathcal{X}}_1^{k+1}=\arg\min_{\boldsymbol{\mathcal{X}}_1}\sum_{n=1}^{k_1}\left\|\boldsymbol{\mathcal{X}}_{1_{::n}}\right\|_{S_p}^p+\frac{\mu_k}{2}\left\|\boldsymbol{\mathcal{X}}_{1_{n:::}}-\boldsymbol{\mathcal{A}}_1^k\right\|_F^2 \tag{28}$$

This can be formulated as the generalized singular value thresholding [60], [61] of each frontal slices of $\boldsymbol{\mathcal{A}}_1^k$, i.e.,

$$\boldsymbol{\mathcal{X}}_{1_{n:::}}^{k+1}=\arg\min_{\boldsymbol{\mathcal{X}}_{1_{n:::}}}\sum_{i=1}^{min(n_2,n_3)}\frac{1}{\mu_k}{\sigma_i}^p(\boldsymbol{\mathcal{X}}_{1_{n:::}})+\frac{1}{2}\left\|\boldsymbol{\mathcal{X}}_{1_{n:::}}-\boldsymbol{\mathcal{A}}_{1_{n:::}}^k\right\|_F^2 \tag{29}$$

where $\boldsymbol{\mathcal{X}}_{1_{n:::}}^{k+1}=\boldsymbol{U}_{1n}^k\boldsymbol{\Sigma}_{1n}^{k+1}\boldsymbol{V}_{1n}^{k\ T}$ and $\boldsymbol{U}_{1n}^k\boldsymbol{S}_{1n}^k\boldsymbol{V}_{1n}^{k\ T}$ is the SVD of $\boldsymbol{\mathcal{A}}_{1_{n:::}}^k$. $\boldsymbol{\Sigma}_{1n}^{k+1}=\mathrm{diag}(\sigma_1,\cdots,\sigma_{min(n_2,n_3)})$ is optimal for the following problem:

$$\sum_{i=1}^{min(n_2,n_3)}\frac{1}{\mu_k}\sigma_i^p+\frac{1}{2}(\sigma_i-\sigma_i(\boldsymbol{\mathcal{A}}_{1_{n:::}}^k))^2 \tag{30}$$

we adapt generalization of soft-thresholding (GST) in [62] to solve (30) effciently, $\boldsymbol{\Sigma}_{1n}^{k+1}=\mathrm{diag}(GST(\mathrm{diag}(\boldsymbol{S}_{1n}^k),\mu_k,p,J))$, the GST is summarized in Algorithm 1.

$$\begin{aligned}\boldsymbol{\mathcal{X}}_2^{k+1}&=\arg\min_{\boldsymbol{\mathcal{X}}_2}\sum_{n=1}^{k_2}\left\|\boldsymbol{\mathcal{X}}_{2_{:n:}}\right\|_{S_p}^p+\frac{\mu_k}{2}\left\|\boldsymbol{\mathcal{X}}_{2_{:n:}}-\boldsymbol{\mathcal{A}}_2^k\right\|_F^2 \\ \boldsymbol{\mathcal{X}}_3^{k+1}&=\arg\min_{\boldsymbol{\mathcal{X}}_3}\sum_{n=1}^{k_3}\left\|\boldsymbol{\mathcal{X}}_{3_{::n}}\right\|_{S_p}^p+\frac{\mu_k}{2}\left\|\boldsymbol{\mathcal{X}}_{3_{::n}}-\boldsymbol{\mathcal{A}}_3^k\right\|_F^2.\end{aligned} \tag{31}$$

The updating of $\boldsymbol{\mathcal{X}}_2^{k+1}$ and $\boldsymbol{\mathcal{X}}_3^{k+1}$ is similar to that of $\boldsymbol{\mathcal{X}}_1^{k+1}$.

(4) **Update** $\boldsymbol{\mathcal{Z}}$

$$\begin{aligned}\boldsymbol{\mathcal{Z}}^{k+1}&=\frac{\boldsymbol{\mathcal{Y}}_1^{k+1}+\boldsymbol{\mathcal{Y}}_2^{k+1}+\boldsymbol{\mathcal{Y}}_3^{k+1}}{3}, \\ &\pi_\Omega(\boldsymbol{\mathcal{Z}}^{k+1})=\pi_\Omega(\boldsymbol{\mathcal{D}}).\end{aligned} \tag{32}$$

(5) **Update the multipliers** $\{\boldsymbol{\mathcal{P}}_i\}_{i=1}^3$

$$\boldsymbol{\mathcal{P}}_i^{k+1}=\boldsymbol{\mathcal{P}}_i^k+\mu_k(\boldsymbol{\mathcal{X}}_i^{k+1}\times_i\boldsymbol{L}_i^{k+1}-\boldsymbol{\mathcal{Y}}_i^{k+1}). \tag{33}$$

The optimization for (18) is summarized in **Algorithm 2**.

TABLE I: RE Comparsion on Two Types of Synthetic Data under Different SRs. For Synthetic Data-1: $n_1 = n_2 = 50, n_3 = 20, r = 5$ (Upper) and For Synthetic Data-2: $n_1 = n_2 = 100, n_3 = 50, r = 10$ (Bottom). The Best RE Values Are Highlighted in Red.

Method \ SR	20%	30%	40%	50%	60%	70%	80%
HaLRTC [18]	0.8964	0.8372	0.7757	0.7085	0.6303	0.549	0.4372
TMac [63]	0.9232	0.8709	0.8203	0,7680	0.7103	0.7312	0.7755
TCTF [37]	2.4466	2.3018	2.1166	1.9429	1.7688	1.5548	1.3055
TRLRF [64]	1.1781	0.9601	0.8128	0.679	0.591	0.4967	0.3997
t-SVD [65]	0.9574	0.889	0.8127	0.7236	0.6303	0.5255	0.4075
FTNN [52]	1.0123	0.9686	0.9234	0.8809	0.8421	0.8049	0.7664
ESP-LRTC [66]	0.9233	0.8821	0.8367	0.7916	0.7425	0.6889	0.6331
KBR-TC [67]	0.9237	0.8812	0.7166	0.5759	0.7413	0.3474	0.2316
OITNN [40]	0.8946	0.8352	0.7718	0.7079	0.6329	0.5476	0.4435
SALTS	0.635	0.2386	6.07e-9	2.20e-11	2.10e-12	3.81e-14	1.43e-15
HaLRTC [18]	0.8946	0.8372	0.775	0.6926	0.613	0.5255	0.4243
TMac [63]	0.9364	0.9099	0.8927	0.8761	0.9246	0.9865	0.7467
TCTF [37]	2.4752	2.2578	2.00	1.7077	1.3845	1.1124	0.9282
TRLRF [64]	0.9513	0.865	0.7839	0.7035	0.6227	0.5400	0.4396
t-SVD [65]	0.9775	0.9166	0.8473	0.7708	0.6836	0.5886	0.4780
FTNN [52]	1.0171	0.9823	0.9455	0.9073	0.8685	0.8281	0.7883
ESP-LRTC [66]	0.9222	0.8817	0.8366	0.7912	0.7426	0.6902	0.6345
KBR-TC [67]	0.9223	0.8352	0.7719	0.6912	0.5821	0.4788	0.3671
OITNN [40]	0.8940	0.8362	0.7742	0.7059	0.6315	0.5466	0.4468
SALTS	0.6196	0.2333	8.29e-12	1.72e-11	1.30e-12	2.74e-14	2.85e-15

TABLE II: The Average Five Trials of Performance Comparison of 10 Competing TC Methods with SR 3%, 5%, 10%, 20% on 32 MSI in **CAVE** Data.

Method	3%				5%				10%				20%				Times/s
	PSNR	SSIM	FSIM	ERGAS	PSNR	SSIM	FSIM	ERGAS	PSNR	SSIM	FSIM	ERGAS	PSNR	SSIM	FSIM	ERGAS	
HaLRTC [18]	20.31	0.6216	0.7767	526.26	24.88	0.7266	0.8300	341.68	29.68	0.8460	0.8930	200.71	34.88	0.9251	0.9460	114.20	13.50
TMac [63]	24.74	0.6585	0.8176	328.05	26.17	0.7260	0.8461	281.90	29.82	0.8537	0.8979	207.33	35.14	0.9299	0.9477	114.74	19.43
TCTF [37]	14.65	0.0481	0.4019	1204.05	20.81	0.4433	0.6803	488.94	22.87	0.5437	0.7370	391.82	24.53	0.6293	0.7926	312.95	53.44
TRLRF [64]	25.99	0.5407	0.7334	292.67	30.81	0.7879	0.8607	159.65	33.37	0.8663	0.9063	116.55	38.01	0.9408	0.9546	69.60	457.23
t-SVD [65]	27.76	0.7506	0.8441	235.52	30.29	0.8202	0.8835	180.58	34.60	0.9046	0.9344	114.21	40.05	0.9616	0.9718	63.83	551.95
FTNN [52]	30.31	0.8453	0.8977	189.35	33.68	0.9087	0.9354	129.18	38.31	0.9590	0.9682	76.45	43.63	0.9847	0.9872	43.01	348.77
ESP-LRTC [66]	30.42	0.8385	0.8871	171.44	35.49	0.9107	0.9381	100.35	41.79	0.9708	0.9776	47.14	47.49	0.9897	0.9921	25.11	190.61
KBR-TC [67]	31.11	0.8437	0.8960	168.10	36.15	0.9272	0.9485	95.64	42.48	0.9761	0.9820	46.42	48.62	0.9928	0.9946	23.23	168.24
OITNN [40]	31.15	0.8832	0.9163	164.01	34.20	0.9280	0.9455	114.90	39.23	0.9692	0.9749	67.23	45.56	0.9894	0.9909	35.50	132.41
SALTS	33.74	0.8974	0.9319	119.80	38.60	0.9579	0.9703	72.04	44.36	0.9862	0.9897	38.58	51.35	0.9971	0.9974	17.05	477.29

Algorithm 1 Generalization of Soft-Thresholding(GST)

Input: $\boldsymbol{s}, \mu, p, J$.

$\tau_p^{GST}(\mu) = (2\mu(1-p))^{\frac{1}{2-p}} + \mu p(2\mu(1-p))^{\frac{p-1}{2-p}}$;
if $|\boldsymbol{s}| \leq \tau_p^{GST}(\mu)$ **then**
  $\boldsymbol{S}_p^{GST}(\boldsymbol{s};\mu) = 0$;
else
  $k = 0, \boldsymbol{x}^{(k)} = \boldsymbol{s}$;
  for $k = 0, 1, \cdots, J$ **do**
    $\boldsymbol{x}^{(k+1)} = \boldsymbol{s} - \mu p(\boldsymbol{x}^{(k)})^{p-1}$;
    $k = k + 1$;
  end for
  $\boldsymbol{S}_p^{GST}(\boldsymbol{s};\mu) = \boldsymbol{x}^{(k)})$;
end if

Output: $\boldsymbol{S}_p^{GST}(\boldsymbol{s};\mu)$.

Algorithm 2 SALTS for tensor completion

Input: Observation samples $\boldsymbol{\mathcal{D}}_{ijk}$ of tensor $\boldsymbol{\mathcal{D}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$,$(i,j,k) \in \Omega$, transform rank k_1, k_2, k_3 .

Initialize: $\epsilon = 1e-7, \mu^0 = 1e-2, \mu_{max} = \rho_{max} = 1e10, \eta = 1.01, \boldsymbol{L}_i{}^0 = \boldsymbol{\mathcal{X}}_i^0 = \boldsymbol{\mathcal{Y}}_i^0 = \boldsymbol{\mathcal{E}}^0 = \boldsymbol{\mathcal{Y}}^0$, Maxiter=800 .
while not convergence and $k \leq$ Maxiter **do**
  $k = k + 1$;
  for $i = 1$; $i \leq 3$; $i++$ **do**
    Update $\boldsymbol{\mathcal{Y}}_i^k$ by (22);
    Update $\boldsymbol{L}_i^k$ by (24);
    Update $\boldsymbol{\mathcal{X}}_i^k$ by (28), (30) and GST in Algorithm 1;
  end for
  Update $\boldsymbol{\mathcal{Z}}^k$ by (32);
  Update the multipliers $\{\boldsymbol{\mathcal{P}}_i^k\}$ by (33);
  Update $\mu^k = \min(\mu_{max}, \eta\mu^{k-1})$;
  Check the convergence conditions: $\left\|\boldsymbol{\mathcal{Z}}^k - \boldsymbol{\mathcal{Z}}^{k-1}\right\|_\infty < \epsilon$
end while

Output: Completion Tensor $\boldsymbol{\mathcal{Z}}$.

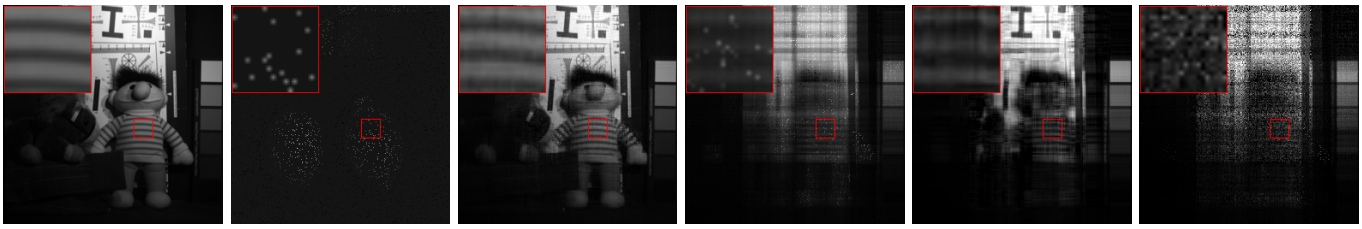
E. Convergence Analysis

Because the SALTS model in (18) is a non-convex and non-smooth optimization problem with more than two variables. As the convergence analysis of the original ADMM algorithm has not been established for non-convex problems or for convex problems with more than two block variables in general cases, it is difficult to determine the global convergence under the framework of ADMM [68], [69]. However in this paper we will prove the weak convergence of Algorithm 2 in Theorem 1 which is essential to guarantee that the iterative sequence can attain stable solution. In addition, the stable convergence of algorithm and efficiency of the proposed method will be validated in the experimental section.

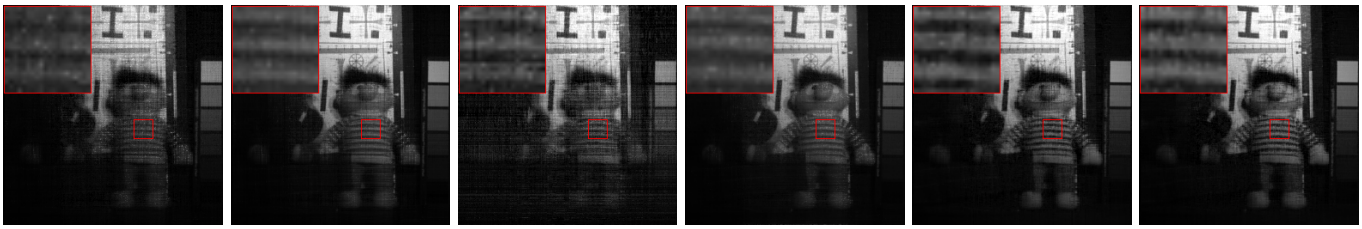
Theorem 1. *Let* $\{\{\boldsymbol{\mathcal{X}}_i^k\}, \{\boldsymbol{L}_i^k\}, \{\boldsymbol{\mathcal{Y}}_i^k\}, \boldsymbol{\mathcal{Z}}^k, \{\boldsymbol{\mathcal{P}}_i^k\}\}_{k=1}^{\infty}$ *be the*

TABLE III: The Average Performance Comparison of Different SRs with 5%, 10%, 20%, 30% for 9 Competing TC Methods on **Akiyo**, **Container**, **Hall**, **Grandma** Color Videos.

Video	metric	HaLRTC [18]	TMac [63]	TRLRF [64]	t-SVD [65]	FTNN [52]	ESP-LRTC [66]	OITNN [40]	KBR-TC [67]	SALTS
Akiyo	PSNR	26.40	31.51	34.07	33.71	35.53	36.76	36.12	37.19	39.13
	SSIM	0.8118	0.8905	0.9185	0.9450	0.9640	0.9615	0.9724	0.9657	0.9832
	FSIM	0.8886	0.9329	0.9534	0.9686	0.9778	0.9779	0.9807	0.9798	0.9889
	ERGAS	129.04	71.79	56.60	56.15	46.53	40.33	44.04	41.10	32.26
Container	PSNR	24.50	29.73	32.39	34.56	34.47	37.49	36.88	37.81	39.08
	SSIM	0.8023	0.8766	0.9159	0.9404	0.9530	0.9579	0.9650	0.9585	0.9736
	FSIM	0.8544	0.9212	0.9512	0.9678	0.9727	0.9786	0.9797	0.9785	0.9867
	ERGAS	134.07	76.87	58.59	47.98	46.32	33.90	38.92	35.34	29.61
Hall	PSNR	24.08	29.46	31.97	32.05	33.71	32.98	33.68	33.38	35.14
	SSIM	0.7790	0.8798	0.9229	0.9385	0.9617	0.9314	0.9637	0.9425	0.9692
	FSIM	0.8450	0.9072	0.9472	0.9604	0.9719	0.9538	0.9729	0.9608	0.9782
	ERGAS	147.84	78.78	61.02	58.59	51.70	53.28	50.00	53.61	43.41
Grandma	PSNR	26.81	31.39	34.78	34.78	36.43	37.07	37.44	37.93	38.96
	SSIM	0.7803	0.8560	0.9209	0.9352	0.9521	0.9426	0.9645	0.9530	0.9696
	FSIM	0.8535	0.9163	0.9518	0.9562	0.9660	0.9665	0.9716	0.9716	0.9787
	ERGAS	152.45	98.79	61.55	58.23	52.30	49.66	46.65	45.66	39.92
Average Time/s		13.10	22.35	500.73	304.64	575.40	180.96	127.17	183.50	599.25



(a) Original image (b) Missed image (c) SALTS (d) HaLRTC (e) TMac (f) TCTF



(g) tSVD (h) FTNN (i) TRLRF (j) OITNN (k) ESP-LRTC (l) KBR-TC

Fig. 3: (a) Original image selected from the band at 700nm of **chart and stuffed toy** in CAVE; (b) The corresponding sampled image with sampling rate 5% percent; (d)-(l) Completing images obtained by 9 competing algorithms. (c) Completing results by proposed SALTS.

sequences generated by Algorithm 2. If $\partial\left\|\boldsymbol{\mathcal{X}}_i^k\right\|_{S_p}^p, \forall i,k$ *is bounded, then we have:*
(a) the sequence $\{\{\boldsymbol{\mathcal{X}}_i^k\},\{\boldsymbol{L}_i^k\},\{\boldsymbol{\mathcal{Y}}_i^k\},\boldsymbol{\mathcal{Z}}^k,\{\boldsymbol{\mathcal{P}}_i^k\}\}_{k=1}^{\infty}$ *is bounded;*
(b) the sequence $\{\{\boldsymbol{\mathcal{X}}_i^k \times_i \boldsymbol{L}_i^k\},\{\boldsymbol{\mathcal{Y}}_i^k\},\boldsymbol{\mathcal{Z}}^k\}_{k=1}^{\infty}$ *is Cauchy sequence.*

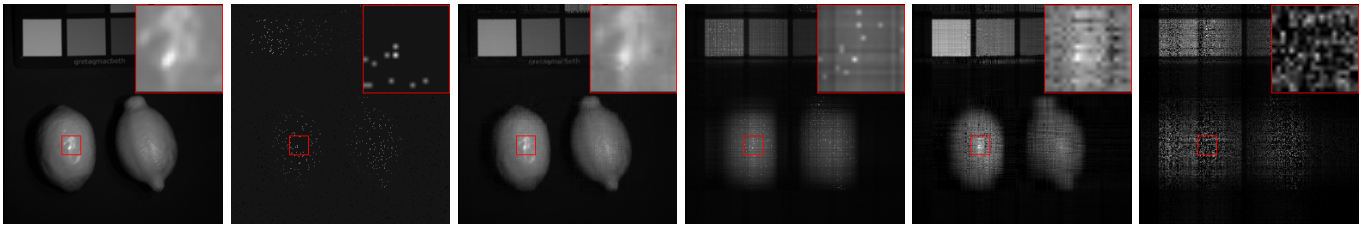
Due to space limits, the detailed proof of **Theorem 1** is in **Supplementary Material**.

F. Computational Complexity

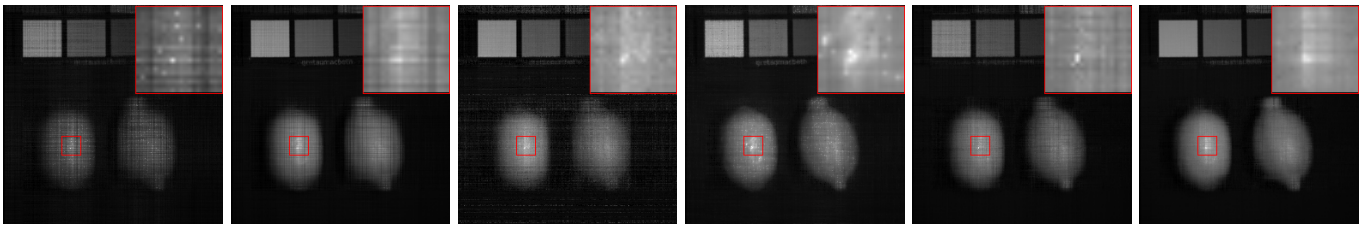
Without loss of generality, for the tensor $\boldsymbol{\mathcal{Y}}$, we assume $n_1 = n_2 = n_3 = n$ and $k_1 = k_2 = k_3 = k$. The main computational complexity of Algorithm 2 is the updating for $\{\boldsymbol{\mathcal{X}}_i\}$, which needs to compute the SVD of multiple matrices. For $\boldsymbol{\mathcal{X}}_i$, computation of SVD is $\mathcal{O}(kn^3)$, which is because there are k slices of $n \times n$ needed to compute full SVD occupying $\mathcal{O}(n^3)$. It is worth mentioning that the time complexity of SVD can be reduced significantly when using partial SVD. The time complexity of SVD occupied by updating $\{\boldsymbol{L}_i\}$ is $\mathcal{O}(k^2n)$. The last computational complexity is multiplication operation of updating $\{\boldsymbol{\mathcal{Y}}_i\}$, which occupies $\mathcal{O}(kn^3)$. Thus the overall time complexity of Algorithm 2 in $\mathcal{O}(6kn^3 + 3nk^2)$. In addition, the space complexity of Algorithm 2 is $\mathcal{O}(n^3)$, owing to storage requirement of $\boldsymbol{\mathcal{X}}_i, \boldsymbol{\mathcal{Y}}_i, \boldsymbol{\mathcal{P}}_i, \boldsymbol{\mathcal{Z}}, \boldsymbol{\mathcal{D}}, \boldsymbol{L}_i, \frac{1}{\alpha}$.

IV. EXPERIMENTAL RESULTS

In order to validate the effectiveness of the proposed SALTS in tensor completion task, we consider the following datasets: Synthetic Data, Hyper-Spectral Image (HSI), Color Video Sequence, Magnetic Resonance Imaging data (MRI), and COIL-20. Due to the limitation of space, the experimental results in verification of robustness , analysis of low rank structure learned, parameters of robustness, convergence verification and more detailed results for proposed SALTS are placed in **Supplementary Material**. We compare proposed methods with nine recently developed algorithms of low rank tensor completion: tSVD [65], FTNN [52] TCTF [37], OITNN [40],



(a) Original image (b) Missed image (c) SALTS (d) HaLRTC (e) TMac (f) TCTF



(g) tSVD (h) FTNN (i) TRLRF (j) OITNN (k) ESP-LRTC (l) KBR-TC

Fig. 4: (a) Original image selected from the band at 700nm of **fake and real lemons** in CAVE; (b) The corresponding sampled image with sampling rate 3%; (d)-(l) Completing images obtained by 9 competing algorithms. (c) Completing results by proposed SALTS.

HaLRTC [18], TMac [63], ESP-LRTC [66], KBR-TC [67], TRLRF [64].

In addition to the intuitive visual display, we apply four quantitative image quality indices used in [67] to evaluate performance of all algorithms numerically, including peak signal-to-noise ratio (PSNR), structure similarity (SSIM), feature similarity (FSIM), erreur relative globale adimensionnelle de synthèse (ERGAS). The larger former three indices, the better recovery performance. On the contrary, smaller ERGAS means better recovery performance. The parameters of all algorithms are well tuned to provide their best performance as possible according to guidance in their reference papers or codes. All experiments are conducted in MATLAB 2020a in Windows 10 on a computer with Intel(R) Core(TM) i9-9900K CPU@3.60GHz and 128GB RAM. In addition, we record the time costs of all methods.

A. Synthetic Data

We generate tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ with lrank r for each frontal slice, which means $\boldsymbol{\mathcal{X}}(:,:,n) = randn(n_1, r) * randn(r, n_2)$ by MATLAB commands. Then we construct the random orthogonal square transform $\boldsymbol{L}_3 \in \mathcal{R}^{n_3 \times n_3}$ along the third dimension. The ground-truth tensor is generated by transformation as $\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{X}} \times_3 \boldsymbol{L}_3$. We test two synthetic data: synthetic data-1 with $n_1 = n_2 = 50, n_3 = 20, r = 5$ and synthetic data-2 with $n_1 = n_2 = 100, n_3 = 50, r = 10$ with different sampling rates (SRs) in $\{20\%, 30\%, 40\%, 50\%, 60\%, 70\%, 80\%\}$. We use the relative error (RE) to quantitatively evaluate the performance of all methods:

$$\text{RE} = \left\|\boldsymbol{\mathcal{Y}} - \hat{\boldsymbol{\mathcal{Y}}}\right\|_F / \left\|\boldsymbol{\mathcal{Y}}\right\|_F \tag{34}$$

where $\boldsymbol{\mathcal{Y}}$ and $\hat{\boldsymbol{\mathcal{Y}}}$ are the original tensor and the recovered tensor, respectively. The RE values are summarized in Table I for synthetic data-1 and synthetic data-2, respectively. We use red color to highlight the best result in each case. Obviously, our method SALTS achieved the minimum RE across all SRs. Even in case of SR $\geq 40\%$, the proposed SALTS could achieve almost exactly recovery, while all other methods fail.

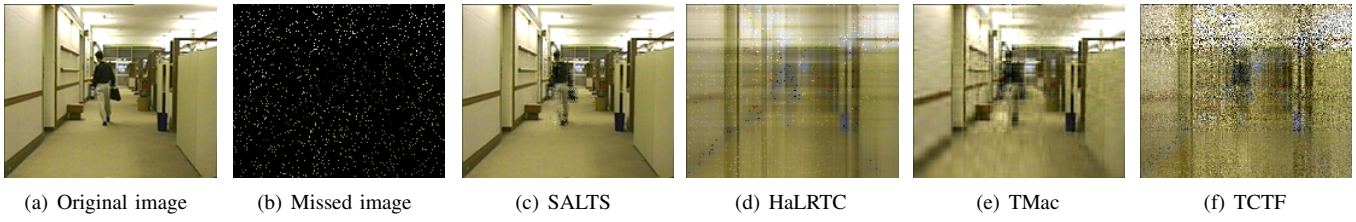
B. HSI Completion

CAVE [1] is a open-source dataset hyperspectral image (HSI) used to test algorithms in many papers. It contains 32 different hyperspectral objects and each size is $512 \times 512 \times 31$, where 512 and 31 denote the spatial resolution and number of spectral bands, respectively. For efficient comparison, we resize each HSI into $256 \times 256 \times 31$ and normalize pixel values into [0,1]. Because all the methods achieve very accurate recovery results when the sampling rate (SR) is high, so we test four different SRs: 3%, 5%, 10% and 20%. The average metric of each tested algorithm for all 32 hyperspectral objects under these four SRs are reported in TABLE II, which shows that our SALTS outperforms all competing methods under all SRs with respect to all metrics. Selected visual results in Fig.4 and Fig.3 from **fake and real lemons** and **chart and stuffed toy** reveal the superiority of the SALTS, both are in the completion of finer-grained textures and coarser-grained structures. The values of four metric PSNR, SSIM, FSIM and ERGAS across all bands in **Supplementary Material**. It can be seen that the proposed SALTS obtains the best metric in all bands.

C. Color Video Completion

We select four color video sequences **Akiyo**, **Hall**, **Container**, and **Grandma** from open-source YUV video dataset[2]. For comparing efficiently, we choose the first 100 frames of each video to test all algorithms. As the color video in

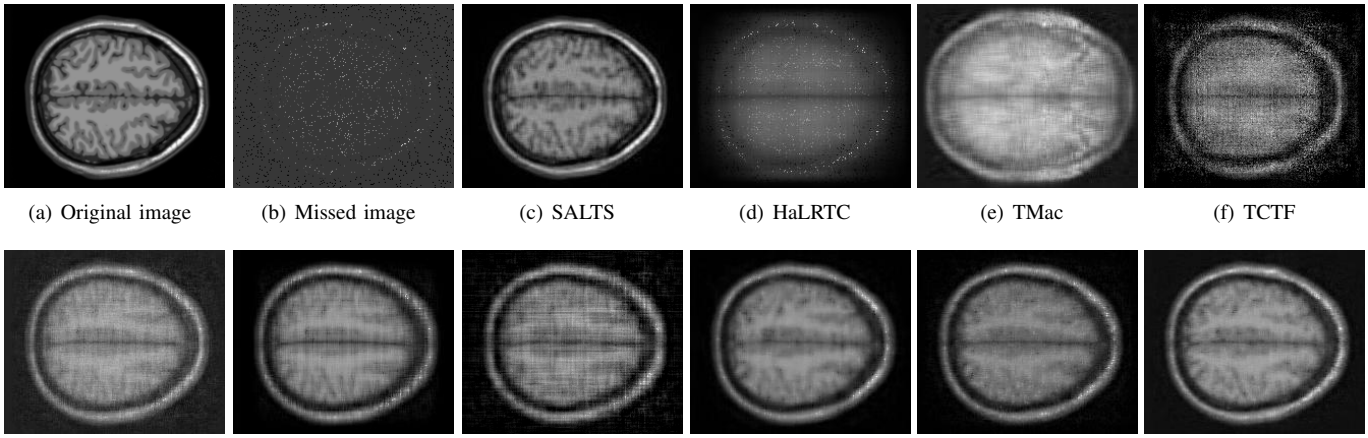
[1] http://www1.cs.columbia.edu/CAVE/databases/multispectral/
[2] http://trace.eas.asu.edu/yuv/



(a) Original image (b) Missed image (c) SALTS (d) HaLRTC (e) TMac (f) TCTF

(g) tSVD (h) FTNN (i) TRLRF (j) OITNN (k) ESP-LRTC (l) KBR-TC

Fig. 5: (a) Original image selected from the 76-th frame of **Hall** color video; (b) The corresponding sampled image with sampling rate 5%; (d)-(l) Completing images obtained by 9 competing algorithms. (c) Completing results by proposed SALTS.



(a) Original image (b) Missed image (c) SALTS (d) HaLRTC (e) TMac (f) TCTF

(g) tSVD (h) FTNN (i) TRLRF (j) OITNN (k) ESP-LRTC (l) KBR-TC

Fig. 6: (a) Original image selected from selected from the 120-th frame of **MRI** data; (b) The corresponding sampled image with sampling rate 5%; (d)-(l) Completing images obtained by 9 competing algorithms. (c) Completing images by our proposed SALTS.

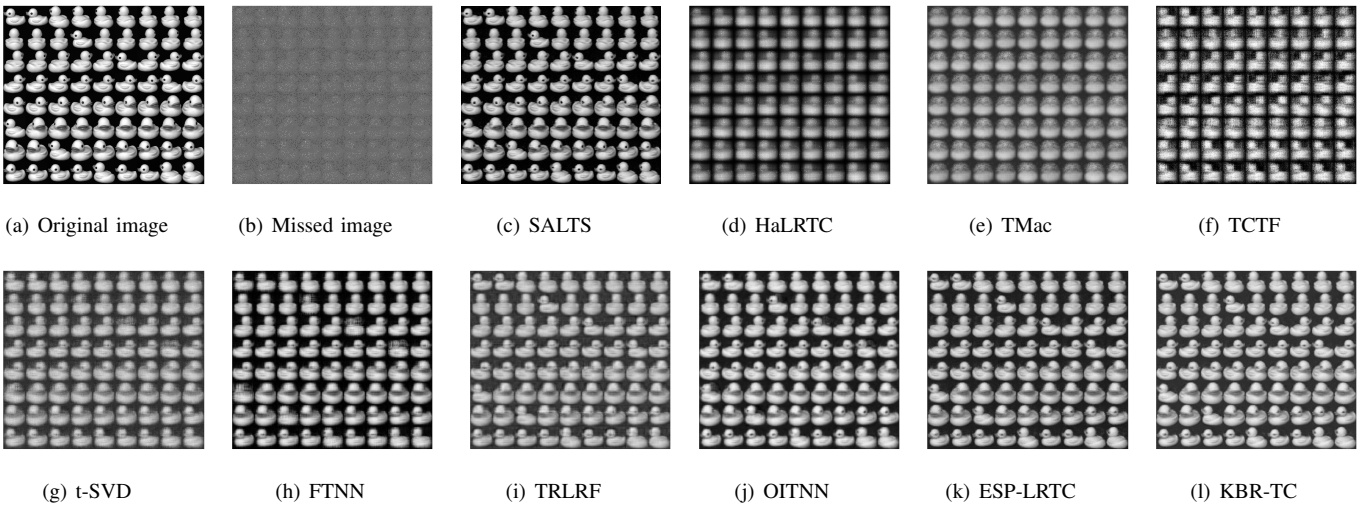
TABLE IV: The Performance Comparison 9 Competing TC Methods with 5%, 10% and 20% on **MRI** Data.

Method	5%				10%				20%				Times/s
	PSNR	SSIM	FSIM	ERGAS	PSNR	SSIM	FSIM	ERGAS	PSNR	SSIM	FSIM	ERGAS	
HaLRTC [18]	16.75	0.2814	0.6283	557.94	19.57	0.4246	0.7182	404.61	23.89	0.6488	0.8246	244.78	46.69
TMac [63]	17.25	0.3348	0.6564	553.39	23.40	0.5891	0.7901	262.30	29.43	0.8041	0.8743	125.91	30.45
TRLRF [64]	22.30	0.4256	0.7289	296.50	25.50	0.5744	0.7922	201.73	29.22	0.7371	0.8623	130.84	302.66
t-SVD [65]	22.40	0.4647	0.7380	308.39	25.22	0.6131	0.7995	222.83	29.29	0.7822	0.8746	139.89	496.36
FTNN [52]	24.02	0.6771	0.8275	250.66	27.08	0.7969	0.8814	171.91	31.27	0.9032	0.9353	104.55	981.15
ESP-LRTC [66]	23.27	0.5484	0.7789	262.21	30.88	0.8634	0.8977	108.51	35.46	0.9456	0.9478	63.73	157.87
KBR-TC [67]	25.82	0.7259	0.8306	195.13	32.17	0.9090	0.9223	92.96	36.57	0.9648	0.9651	55.65	530.90
OITNN [40]	25.00	0.6951	0.8218	217.65	28.47	0.8286	0.8838	144.80	32.85	0.9244	0.9383	87.23	356.18
SALTS	29.55	0.8304	0.8985	124.88	33.18	0.9214	0.9432	82.38	36.97	0.9657	0.9724	53.24	1806.72

RGB format is fourth-order tensor $144 \times 176 \times 3 \times 100$, we convert RGB video into 3D video $144 \times 176 \times 100$ in YUV format as the input of all algorithms and finally the visual results are transferred back to RGB format for presentation. Table III and Fig.5 show that the performance of our proposed SALTS in color video completion is superior to compared methods in both metric and intuitive visual results. Especially, the dynamic foreground of the moving persons in videos is restored more clearly and completely by the proposed algorithm. To further validate the effectiveness of the proposed method, we plot the values of metric across all the frames on **Akiyo**, **Hall** in **Supplementary Material**. This is easy to observe that the proposed SALTS obtains the best metric almost in all frames of tested color videos.

TABLE V: The Average Performance Comparison of 10 Competing TC Methods with 10%, 20%, 30% on 3 Objects in **COIL-20** Data.

Method	10%				20%				30%				Time/s
	PSNR	SSIM	FSIM	ERGAS	PSNR	SSIM	FSIM	ERGAS	PSNR	SSIM	FSIM	ERGAS	
HaLRTC [18]	17.89	0.5196	0.7294	382.84	21.55	0.6625	0.8112	256.67	24.11	0.7852	0.8646	194.48	15.96
TMac [63]	18.24	0.5253	0.7352	369.85	22.98	0.6584	0.7596	217.86	25.46	0.7549	0.8459	163.80	11.37
TRLRF [64]	23.37	0.7420	0.8393	212.66	26.19	0.8353	0.8936	155.80	28.28	0.8861	0.9250	123.43	186.67
t-SVD [65]	20.10	0.4934	0.7222	302.47	22.84	0.6536	0.7933	222.70	25.03	0.7213	0.8462	175.06	62.97
FTNN [52]	22.20	0.6981	0.8165	244.35	25.06	0.8096	0.8777	177.44	27.22	0.8724	0.9146	139.47	185.81
ESP-LRTC [66]	24.81	0.7320	0.8408	184.98	27.26	0.7999	0.8799	141.88	29.28	0.8569	0.9140	113.49	79.29
KBR-TC [67]	25.33	0.7570	0.8525	173.64	27.86	0.8309	0.8968	132.82	30.10	0.8923	0.9321	103.79	92.70
OITNN [40]	23.37	0.7420	0.8393	212.66	26.19	0.8348	0.8936	155.80	28.28	0.8861	0.9250	123.43	41.34
SALTS	25.93	0.8025	0.8829	161.32	28.66	0.8837	0.9281	119.84	30.70	0.9221	0.9514	96.11	477.37



(a) Original image (b) Missed image (c) SALTS (d) HaLRTC (e) TMac (f) TCTF

(g) t-SVD (h) FTNN (i) TRLRF (j) OITNN (k) ESP-LRTC (l) KBR-TC

Fig. 7: (a) One object's original images from 721-th frame to 792-th frame of **COIL-20** ; (b) The sampled images with sampling rate 5%; (d)-(l) Completing results obtained by 9 competing algorithms. (c) Completing results by proposed SALTS.

D. Magnetic Resonance Imaging (MRI) Completion

In this subsection, we use the **MRI** data with volume $181 \times 217 \times 181$ from Simulated Brain Database[3], which contains a set of realistic MRI data volumes produced by an MRI simulator used by the neuroimaging community widely. We set SR as 10%, 20%, and 30% for **MRI**. Table IV and Fig.6 show the quantitative assessments and recovered results of all methods for the 120-th slice, respectively. It is observed from visual images that not only can the contour of missed MRI be best restored, but also the details of the images are inpainted with more fullness and integrity by our proposed method. In order to fully verify the recovery effect in all frames of **MRI**, we plot the four metrics across all frames for all competitive methods in **Supplementary Material**. In addition, our algorithm achieves much superior performance of restoration on all frames in all metrics.

E. COIL-20 Data Completion

In this subsection, we adapt COIL-20[4], which includes 1440 various images of different 20 objects taken from different angles and the size of each image is 128×128. We select 3 different objects from 20 objects with all 72 pictures of each from different angles, which means that each selected object's tensor data size is $128 \times 128 \times 72$. We set SR as 10%, 20%, 30% for each object, respectively. For visual comparison, we set the SR as 5% for **object 2** with comparison of all methods and the corresponding visual, average metric of three objects for different SRs and metrics across all angles for **object 2** are reported in Fig.7, TABLE V and **Supplementary Material**, respectively. In comparison, the proposed algorithm obtains the optimal restoration result from all aspects of the experimental verification.

[3]http://brainweb.bic.mni.mcgill.ca/brainweb/

[4]http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.PHP

V. Conclusion

In this paper, we have proposed a novel multiple self-adaptive learnable transforms (SALTS) based tensor nuclear norm and developed an efficient ADMM to update multiple learnable transforms and transformed tensors along all modes alternately. In the proposed framework, SALTS are jointly learned with transformed tensors by minimizing its induced Schatten-p quasi norms, which are data-dependent and flexibly adaptable to a wide range of data. Due to the adaptation of SALTS, its insensitivity of orientation guarantees sufficient low rank structure of transformed tensors along multiple modes. The visual and numerical results on four benchmark datasets have shown the superiority of the proposed SALTS. In future work, we will apply our SALTS to tensor robust

principal component analysis, which is another fundamental topic in tensor data processing.

ACKNOWLEDGEMENT

The work was supported by National Natural Science Foundation of China (No. 61971093, No. 61960206010, No. 61527803). The work was supported by NSFC Projects of International Cooperation and Exchanges No. 61960206010, supported by the International Science and Technology Innovation Cooperation Project of Sichuan Province: 2021YFH0036, and Fundamental Research Funds for the Central Universities (Grant No. ZYGX2019J067). This work was also partially supported by the research funding T00120210002 of Shenzhen Research Institute of Big Data and the Youth program 62106211 of the National Natural Science Foundation of China.

REFERENCES

[1] Z. Lin and H. Zhang, *Low-Rank Models in Visual Analysis: Theories, Algorithms, and Applications*, 01 2017.

[2] N. D. Sidiropoulos, L. D. Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE Trans. Signal Process.*, vol. 65, no. 13, pp. 3551–3582, 2017.

[3] A. Cichocki, D. P. Mandic, L. D. Lathauwer, G. Zhou, Q. Zhao, C. F. Caiafa, and A. H. Phan, "Tensor decompositions for signal processing applications: From two-way to multiway component analysis," *IEEE Signal Process. Mag.*, vol. 32, no. 2, pp. 145–163, 2015.

[4] J. A. Bengua, H. N. Phien, H. D. Tuan, and M. N. Do, "Efficient tensor completion for color image and video recovery: Low-rank tensor train," *IEEE Trans. Image Process.*, vol. 26, no. 5, pp. 2466–2479, 2017.

[5] W. Hu, D. Tao, W. Zhang, Y. Xie, and Y. Yang, "The twist tensor nuclear norm for video completion," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 28, no. 12, pp. 2961–2973, 2017.

[6] W. Dong, T. Huang, G. Shi, Y. Ma, and X. Li, "Robust tensor approximation with laplacian scale mixture modeling for multiframe image and video denoising," *IEEE J. Sel. Top. Signal Process.*, vol. 12, no. 6, pp. 1435–1448, 2018.

[7] I. Kajo, N. S. Kamel, and Y. Ruichek, "Incremental tensor-based completion method for detection of stationary foreground objects," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 5, pp. 1325–1338, 2019.

[8] A. Sobral and E. Zahzah, "Matrix and tensor completion algorithms for background model initialization: A comparative evaluation," *Pattern Recognit. Lett.*, vol. 96, pp. 22–33, 2017.

[9] I. Kajo, N. S. Kamel, Y. Ruichek, and A. S. Malik, "Svd-based tensor-completion technique for background initialization," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 3114–3126, 2018.

[10] L. Zhuang and J. M. Bioucas-Dias, "Fast hyperspectral image denoising and inpainting based on low-rank and sparse representations," *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.*, vol. 11, no. 3, pp. 730–742, 2018.

[11] Y. Chang, L. Yan, X. Zhao, H. Fang, Z. Zhang, and S. Zhong, "Weighted low-rank tensor recovery for hyperspectral image restoration," *IEEE Trans. Cybern.*, vol. 50, no. 11, pp. 4558–4572, 2020.

[12] J. Xue, Y. Zhao, Y. Bu, J. C.-W. Chan, and S. G. Kong, "When laplacian scale mixture meets three-layer transform: A parametric tensor sparsity for tensor completion," *IEEE Transactions on Cybernetics*, pp. 1–15, 2022.

[13] J. Xue, Y. Zhao, S. Huang, W. Liao, J. C.-W. Chan, and S. G. Kong, "Multilayer sparsity-based tensor decomposition for low-rank tensor completion," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2021.

[14] D. Shah and C. L. Yu, "Iterative collaborative filtering for sparse noisy tensor estimation," in *IEEE International Symposium on Information Theory, ISIT 2019, Paris, France, July 7-12, 2019*. IEEE, 2019, pp. 41–45.

[15] X. Bi, A. Qu, and X. Shen, "Multilayer tensor factorization with applications to recommender systems," *Annals of Statistics*, 2017.

[16] J. Wu, Z. Lin, and H. Zha, "Essential tensor learning for multi-view spectral clustering," *IEEE Trans. Image Process.*, vol. 28, no. 12, pp. 5910–5922, 2019.

[17] P. Zhou, C. Lu, J. Feng, Z. Lin, and S. Yan, "Tensor low-rank representation for data recovery and clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[18] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 208–220, 2013.

[19] L. Lim and P. Comon, "Blind multilinear identification," *IEEE Trans. Inf. Theory*, vol. 60, no. 2, pp. 1260–1280, 2014.

[20] S. Gandy, B. Recht, and I. Yamada, "Tensor completion and low—rank tensor recovery via convex optimization," *Inverse Problems*, vol. 27, p. 025010, 01 2011.

[21] H. Kiers, "Towards a standardized notation and terminology in multiway analysis," *Journal of Chemometrics - J CHEMOMETR*, vol. 14, pp. 105–122, 05 2000.

[22] L. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, no. 3, pp. 279–311, 1966.

[23] J. Håstad, "Tensor rank is np-complete," *J. Algorithms*, vol. 11, no. 4, pp. 644–654, 1990.

[24] C. J. Hillar and L. Lim, "Most tensor problems are np-hard," *J. ACM*, vol. 60, no. 6, pp. 45:1–45:39, 2013.

[25] Misha, E., Kilmer, Carla, D., and Martin, "Factorization strategies for third-order tensors - sciencedirect," *Linear Algebra & Its Applications*, vol. 435, no. 3, pp. 641–658, 2011.

[26] K. Braman, "Third-order tensors as linear operators on a space of matrices," *Linear Algebra and its Applications*, vol. 433, no. 7, pp. 1241–1253, 2010.

[27] M. E. Kilmer, K. S. Braman, N. Hao, and R. C. Hoover, "Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging," *SIAM J. Matrix Anal. Appl.*, vol. 34, no. 1, pp. 148–172, 2013.

[28] E. Kernfeld, M. Kilmer, and S. Aeron, "Tensor–tensor products with invertible linear transforms," *Linear Algebra & Its Applications*, vol. 485, pp. 545–570, 2015.

[29] Z. Zhang and S. Aeron, "Exact tensor completion using t-svd," *IEEE Trans. Signal Process.*, vol. 65, no. 6, pp. 1511–1526, 2017.

[30] Z. Zhang, G. Ely, S. Aeron, N. Hao, and M. E. Kilmer, "Novel methods for multilinear data completion and de-noising based on tensor-svd," pp. 3842–3849. [Online]. Available: https://doi.org/10.1109/CVPR.2014.485

[31] 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 Anal. Mach. Intell.*, vol. 42, no. 4, pp. 925–938, 2020.

[32] Y. Mu, P. Wang, L. Lu, X. Zhang, and L. Qi, "Weighted tensor nuclear norm minimization for tensor completion using tensor-svd," *Pattern Recognit. Lett.*, vol. 130, pp. 4–11, 2020.

[33] T. Jiang, T. Huang, X. Zhao, and L. Deng, "Multi-dimensional imaging data recovery via minimizing the partial sum of tubal nuclear norm," *J. Comput. Appl. Math.*, vol. 372, p. 112680, 2020.

[34] H. Kong, X. Xie, and Z. Lin, "t-schatten-p norm for low-rank tensor recovery," *IEEE J. Sel. Top. Signal Process.*, vol. 12, no. 6, pp. 1405–1419, 2018.

[35] C. Liu, H. Shan, and C. Chen, "Tensor p-shrinkage nuclear norm for low-rank tensor completion," *Neurocomputing*, vol. 387, pp. 255–267, 2020.

[36] J. Lou and Y. Cheung, "Robust low-rank tensor minimization via a new tensor spectral k-support norm," *IEEE Trans. Image Process.*, vol. 29, pp. 2314–2327, 2020.

[37] P. Zhou, C. Lu, Z. Lin, and C. Zhang, "Tensor factorization for low-rank tensor completion," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1152–1163, 2018.

[38] Y. Zhou and Y. Cheung, "Bayesian low-tubal-rank robust tensor factorization with multi-rank determination," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 62–76, 2021.

[39] X. Liu, S. Aeron, V. Aggarwal, and X. Wang, "Low-tubal-rank tensor completion using alternating minimization," *IEEE Trans. Inf. Theory*, vol. 66, no. 3, pp. 1714–1737, 2020.

[40] A. Wang, C. Li, Z. Jin, and Q. Zhao, "Robust tensor decomposition via orientation invariant tubal nuclear norms." AAAI Press, 2020, pp. 6102–6109.

[41] C. Lu, X. Peng, and Y. Wei, "Low-rank tensor completion with a new tensor nuclear norm induced by invertible linear transforms," 2019, pp. 5996–6004.

[42] A. Wang, X. Song, X. Wu, Z. Lai, and Z. Jin, "Generalized dantzig selector for low-tubal-rank tensor recovery," in *ICASSP 2019 - 2019 IEEE*

International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019, pp. 3427–3431.
[43] J. Hou, F. Zhang, H. Qiu, J. Wang, Y. Wang, and D. Meng, "Robust low-tubal-rank tensor recovery from binary measurements," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021.
[44] Q. Jiang and M. Ng, "Robust low-tubal-rank tensor completion via convex optimization," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, S. Kraus, Ed. ijcai.org, 2019, pp. 2649–2655.
[45] W. Sun, Y. Chen, L. Huang, and H. C. So, "Tensor completion via generalized tensor tubal rank minimization using general unfolding," *IEEE Signal Processing Letters*, vol. 25, no. 6, pp. 868–872, 2018.
[46] F. Zhang, J. Hou, J. Wang, and W. Wang, "Uniqueness guarantee of solutions of tensor tubal-rank minimization problem," *IEEE Signal Processing Letters*, vol. 27, pp. 540–544, 2020.
[47] H. Wang, F. Zhang, J. Wang, and Y. Wang, "Estimating structural missing values via low-tubal-rank tensor completion," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 3297–3301.
[48] P. Wang, L. Li, and G. Cheng, "Low rank tensor completion with sparse regularization in a transformed domain," *CoRR*, vol. abs/1911.08082, 2019. [Online]. Available: http://arxiv.org/abs/1911.08082
[49] W. Xu, X. Zhao, and M. K. Ng, "A fast algorithm for cosine transform based tensor singular value decomposition," *CoRR*, vol. abs/1902.03070, 2019. [Online]. Available: http://arxiv.org/abs/1902.03070
[50] C. Lu and P. Zhou, "Exact recovery of tensor robust principal component analysis under linear transforms," *CoRR*, vol. abs/1907.08288, 2019. [Online]. Available: http://arxiv.org/abs/1907.08288
[51] G. Song, M. K. Ng, and X. Zhang, "Robust tensor completion using transformed tensor singular value decomposition," *Numer. Linear Algebra Appl.*, vol. 27, no. 3, 2020.
[52] T. Jiang, M. K. Ng, X. Zhao, and T. Huang, "Framelet representation of tensor nuclear norm for third-order tensor completion," *IEEE Trans. Image Process.*, vol. 29, pp. 7233–7244, 2020. [Online]. Available: https://doi.org/10.1109/TIP.2020.3000349
[53] T.-X. Jiang, X.-L. Zhao, H. Zhang, and M. K. Ng, "Dictionary learning with low-rank coding coefficients for tensor completion," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2021.
[54] H. Kong, C. Lu, and Z. Lin, "Tensor q-rank: New data dependent tensor rank," *Machine Learning*, 2021.
[55] J. Xue, Y. Zhao, W. Liao, and J. C. Chan, "Nonconvex tensor rank minimization and its applications to tensor recovery," *Inf. Sci.*, vol. 503, pp. 109–128, 2019.
[56] X. Zhang, "A nonconvex relaxation approach to low-rank tensor completion," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 30, no. 6, pp. 1659–1671, 2019.
[57] Y. Su, X. Wu, and W. Liu, "Low-rank tensor completion by sum of tensor nuclear norm minimization," *IEEE Access*, vol. 7, pp. 134 943–134 953, 2019. [Online]. Available: https://doi.org/10.1109/ACCESS.2019.2940664
[58] J. Peng, Q. Xie, Q. Zhao, Y. Wang, Y. Leung, and D. Meng, "Enhanced 3dtv regularization and its applications on HSI denoising and compressed sensing," *IEEE Trans. Image Process.*, vol. 29, pp. 7889–7903, 2020.
[59] G. A. Watson, "The solution of orthogonal procrustes problems for a family of orthogonally invariant norms," *Adv. Comput. Math.*, vol. 2, no. 4, pp. 393–405, 1994.
[60] C. Lu, C. Zhu, C. Xu, S. Yan, and Z. Lin, "Generalized singular value thresholding," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, B. Bonet and S. Koenig, Eds. AAAI Press, 2015, pp. 1805–1811.
[61] Y. Xie, S. Gu, Y. Liu, W. Zuo, W. Zhang, and L. Zhang, "Weighted schatten p-norm minimization for image denoising and background subtraction," *IEEE Trans. Image Process.*, vol. 25, no. 10, pp. 4842–4857, 2016.
[62] W. Zuo, D. Meng, L. Zhang, X. Feng, and D. Zhang, "A generalized iterated shrinkage algorithm for non-convex sparse coding," in *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*, 2013, pp. 217–224.
[63] Y. Xu, R. Hao, W. Yin, and Z. Su, "Parallel matrix factorization for low-rank tensor completion," *Inverse Problems & Imaging*, vol. 9, no. 2, pp. 601–624, 2017.
[64] L. Yuan, C. Li, D. P. Mandic, J. Cao, and Q. Zhao, "Tensor ring decomposition with rank minimization on latent space: An efficient approach for tensor completion." AAAI Press, 2019, pp. 9151–9158.
[65] C. Lu, J. Feng, Z. Lin, and S. Yan, "Exact low tubal rank tensor recovery from gaussian measurements." ijcai.org, 2018, pp. 2504–2510.
[66] J. Xue, Y. Zhao, W. Liao, J. C. Chan, and S. G. Kong, "Enhanced sparsity prior model for low-rank tensor completion," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 31, no. 11, pp. 4567–4581, 2020.
[67] Q. Xie, Q. Zhao, D. Meng, and Z. Xu, "Kronecker-basis-representation based tensor sparsity and its applications to tensor recovery," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 8, pp. 1888–1902, 2018.
[68] C. Chen, B. He, Y. Ye, and X. Yuan, "The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent," *Math. Program.*, vol. 155, no. 1-2, pp. 57–79, 2016.
[69] S. P. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.