

Non-Convex Tensor Recovery from Local Measurements

Tongle Wu¹, Ying Sun^{1*}, Jicong Fan^{2*}

¹School of Electrical Engineering and Computer Science, Pennsylvania State University

²School of Data Science, The Chinese University of Hong Kong, Shenzhen
{tfw5381, ybs5190}@psu.edu, fanjicong@cuhk.edu.cn

Abstract

Motivated by the settings where sensing the entire tensor is infeasible, this paper proposes a novel tensor compressed sensing model, where measurements are only obtained from sensing each lateral slice via mutually independent matrices. Leveraging the low tubal rank structure, we reparameterize the unknown tensor \mathcal{X}^* using two compact tensor factors and formulate the recovery problem as a nonconvex minimization problem. To solve the problem, we first propose an alternating minimization algorithm, termed Alt-PGD-Min, that iteratively optimizes the two factors using a projected gradient descent and an exact minimization step, respectively. Despite nonconvexity, we prove that Alt-PGD-Min achieves ϵ -accuracy recovery with $\mathcal{O}(\kappa^2 \log \frac{1}{\epsilon})$ iteration complexity and $\mathcal{O}(\kappa^6 r n_3 \log n_3 (\kappa^2 r (n_1 + n_2) + n_1 \log \frac{1}{\epsilon}))$ sample complexity, where κ denotes tensor condition number of \mathcal{X}^* . To further accelerate the convergence, especially when the tensor is ill-conditioned with large κ , we prove Alt-ScalePGD-Min that preconditions the gradient update using an approximate Hessian that can be computed efficiently. We show that Alt-ScalePGD-Min achieves κ independent iteration complexity $\mathcal{O}(\log \frac{1}{\epsilon})$ and improves the sample complexity to $\mathcal{O}(\kappa^4 r n_3 \log n_3 (\kappa^4 r (n_1 + n_2) + n_1 \log \frac{1}{\epsilon}))$. Experiments validate the effectiveness of the proposed methods.

Introduction

Motivated by the well-known compressed sensing and matrix sensing (Candès, Romberg, and Tao 2006; Recht, Fazel, and Parrilo 2010) problems, tensor compressed sensing (TCS) has attracted increasing attention in recent years (Shi et al. 2013; Rauhut, Schneider, and Stojanac 2017; Tong et al. 2022b; Chen, Raskutti, and Yuan 2019). The goal of TCS is to recover a tensor $\mathcal{X}^* \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ from a few measurements $\mathbf{y} \in \mathbb{R}^m$ ($m \ll n_1 n_2 n_3$), where $\mathbf{y} = \mathcal{A}(\mathcal{X}^*)$ and $\mathcal{A} : \mathbb{R}^{n_1 \times n_2 \times n_3} \rightarrow \mathbb{R}^m$ is a linear operator. Since this problem is ill-posed for arbitrary \mathcal{X}^* , the success of TCS relies on the existence of the low dimensional intrinsic structure in the original high-order tensor \mathcal{X}^* . Such a structure has been widely validated and utilized in many real-world applications such as dynamic MRI (Yu et al. 2014; Gong and Zhang 2024), video compression (Li et al. 2022; Ma

et al. 2019; Liu et al. 2023b), snapshot compressive imaging (Ma et al. 2019; Liu et al. 2023b), quantum computing (Ran et al. 2020; Kuzmin et al. 2024), image recovery (Fan et al. 2023), and collaborative filtering (Fan 2022).

In the literature, different tensor decompositions can induce different definitions of tensor rank (Kolda and Bader 2009), which are often more complex than matrix rank. Consequently, the non-uniqueness and complexity of the tensor rank make TCS a non-trivial extension of matrix sensing. Most works of TCS assume that the ground truth tensor \mathcal{X}^* has a low Tucker rank (Han, Willett, and Zhang 2022; Luo and Zhang 2023; Ahmed, Raja, and Bajwa 2020; Mu et al. 2014) or a low tubal rank (Zhang et al. 2020b; Hou et al. 2021; Lu et al. 2018; Liu et al. 2023a), which are induced by Tucker decomposition (Tucker 1966) and tensor Singular Value Decomposition (t-SVD) (Kilmer and Martin 2011) respectively. In these works, the measurements are given by

$$y_i = \langle \mathcal{A}_i, \mathcal{X}^* \rangle, \quad i \in [m], \quad (1)$$

where $\mathcal{A}_i \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ has i.i.d. zero-mean Gaussian entries and can sense entire \mathcal{X}^* .

It should be pointed out that in many scenarios, it is difficult to sense the entire tensor \mathcal{X}^* , preventing the application of the sensing model (1). For instance, due to memory or privacy limitations, large-scale tensor data may be partitioned into multiple smaller tensors stored in a distributed network (Moothedath and Vaswani 2024; Singh and Vaswani 2024; Wu and Sun 2024). Another example is when the tensor \mathcal{X}^* , such as images or videos, is collected in an on-the-fly streaming setting (Srinivasa et al. 2019).

To address the challenge, we propose a novel tensor compressed sensing model where each measurement is generated from locally sensing a slice of \mathcal{X}^* . Our model is detailed as follows.

Definition 1 (Local TCS) For each lateral slice $i \in [n_2]$, its j -th local measurement y_{ji} is obtained by

$$y_{ji} = \langle \mathcal{A}_i(:, j, :), \mathcal{X}^*(:, i, :) \rangle, \quad i \in [n_2], j \in [m], \quad (2)$$

where the $\mathcal{A}_i(:, j, :) \in \mathbb{R}^{n_1 \times n_3}$ denotes the j -th sensing matrix for i -th lateral slice of \mathcal{X}^* .

Take the dynamic video sensing mentioned before as an example. Under the local TCS model in (2), the entire video is modeled as \mathcal{X}^* , with each lateral slice $\mathcal{X}^*(:, i, :)$ representing the video frame at i -th timestamp (Li, Ye, and Xu

*Corresponding authors.

2017; Zhou et al. 2017; Wang et al. 2020). The observations $\{y_{ji}\}_{j=1}^m$ is obtained by measuring the i -th frame of video \mathcal{X}^* . We aim to recover $\mathcal{X}^* \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ from the measurements $\{y_{ji}\}_{i,j=1}^{i=n_2, j=m}$ obtained via (2). Under this framework, fundamental questions to understand are:

Under what conditions can we provably recover \mathcal{X}^ , and how to compute the solution efficiently?*

This paper considers the problem above under the structural assumption that the ground truth \mathcal{X}^* is of low tubal rank with $r \ll \min\{n_1, n_2, n_3\}$ (see Definition 8). We focus on low tubal rank for two key reasons. First, it can be computed more efficiently by solving multiple SVDs in the Fourier domain compared to CP and Tucker ranks (Zhang et al. 2014). Second, the convolution operator in this model is particularly effective at capturing the ‘‘spatial-shifting’’ properties of data (Liu et al. 2019; Wu et al. 2022; Wu and Fan 2024). Our main contributions are summarized as follows.

- We introduce a novel local TCS model (2) for tensor compressed sensing with measurements obtained by lateral slice-wise sensing. Compared to the traditional TCS model (1), local sensing does not rely on the availability of the entire tensor, which greatly enlarges its applicable scenarios, such as real-time and distributed processing.
- We formulate the recovery problem as a nonconvex minimization problem based on the low tubal rank tensor factorization for \mathcal{X}^* . An alternating minimization algorithm, called Alt-PGD-Min, is proposed to solve the problem with efficient computations per iteration. We show that under suitable conditions on the sensing operator, with $\mathcal{O}(\kappa^6 r n_3 \log n_3 (\kappa^2 r (n_1 + n_2) + n_1 \log \frac{1}{\epsilon}))$ samples Alt-PGD-Min computes a solution that is ϵ -close to \mathcal{X}^* in $\mathcal{O}(\kappa^2 \log \frac{1}{\epsilon})$ iterations, where κ is the tensor condition number of \mathcal{X}^* .
- To improve the dependency of both the sample and iteration complexity on κ , we further proposed Alt-ScalePGD-Min that preconditions the gradient step in Alt-PGD-Min using an approximation of the Hessian matrix that is cheap to compute. We show that by incorporating the preconditioner, Alt-ScalePGD-Min iteration complexity $\mathcal{O}(\log \frac{1}{\epsilon})$ that is independent of κ , and improves the sample complexity to $\mathcal{O}(\kappa^4 r n_3 \log n_3 (\kappa^4 r (n_1 + n_2) + n_1 \log \frac{1}{\epsilon}))$.
- We validated the proposed local sensing model and algorithms on both synthetic and real-world data. Numerical results show that the proposed algorithms can achieve effective performance in the local TCS model (2).

Related Work

Tensor Compressed Sensing (TCS). Canonical TCS problems (1) based on Tucker and t-SVD decompositions have been extensively investigated in recent years. Studies in (Shi et al. 2013; Rauhut, Schneider, and Stojanac 2017; Ahmed, Raja, and Bajwa 2020; Mu et al. 2014; Chen, Raskutti, and Yuan 2019; Han, Willett, and Zhang 2022; Luo and Zhang 2023) utilized convex or non-convex optimization methods to solve low Tucker rank based tensor CS. For

TCS with low tubal rank under the t-SVD framework, Lu et al. (2018) proposed a convex method that minimizes the tensor nuclear norm (TNN) with order optimal sample complexity. Zhang et al. (2020b) proposed a regularized TNN minimization method with provable robust recovery performance from noisy observations based on the defined tensor Restricted Isometry Property (RIP). Hou et al. (2021) proposed convex methods to solve one-bit TCS from binary observations and provided robust recovery guarantees. Liu et al. (2024) developed theoretical guarantees for the non-convex gradient descent method, which deals with exact low tubal rank and overparameterized tensor factorizations for solving the model (1). Liu et al. (2023a) fused low-rankness and local-smoothness of real-world tensor data in TCS and obtained the provable enhanced recovery guarantee. However, all existing TCS studies focus on models where random measurement tensors have access to the entire ground truth tensors, rather than recovering the low-rank tensor through local measurements as proposed in our TCS model in (2).

CS from Local Measurements. Compared to the canonical CS model, the investigation of CS that recovers from local measurements as model (2) is less explored. Nayer and Vaswani (2022); Vaswani (2024); Srinivasa et al. (2019); Srinivasa, Kim, and Lee (2023); Lee et al. (2023) considered matrix sensing model that involves recovering a low-rank matrix from independent compressed measurements of each of its columns. Srinivasa et al. (2019); Srinivasa, Kim, and Lee (2023); Lee et al. (2023) proposed convex programming methods that minimize relevant mixed norms with provable guarantees, while Nayer and Vaswani (2022); Vaswani (2024) proposed efficient non-convex method and obtained improved iteration and sample complexities.

Nevertheless, to our knowledge, all existing studies on CS from local measurements focused solely on two-dimensional matrices, rather than higher-order tensors that widely exist in science and engineering. Reshaping a tensor into a matrix format to apply matrix methods overlooks the interactions across all dimensions and destroys the inherent structures of data. Therefore, it is essential to study TCS from local measurements, as proposed in our model (2), which has not been addressed in the literature, despite its significant applications in areas like video compression for online or distributed settings (Wu and Sun 2024; Srinivasa et al. 2019). Our experimental results such as Figures 1 and 4 will demonstrate the superiority of tensor CS over matrix CS.

Preliminaries

Notations

We use letters $x, \mathbf{x}, \mathbf{X}, \mathcal{X}$ to denote scalars, vectors, matrices, and tensors, respectively. Let $\{a_n, b_n\}_{n \geq 1}$ be any two positive series. We write $a_n \gtrsim b_n$ (or $a_n \lesssim b_n$) if there exists a universal constant $c > 0$ such that $a_n \geq cb_n$ (or $a_n \leq cb_n$). The notations of $a_n = \Omega(b_n)$ and $a_n = \mathcal{O}(b_n)$ share the same meaning with $a_n \gtrsim b_n$ and $a_n \lesssim b_n$.

The i -th horizontal, lateral, and frontal slice matrix of \mathcal{X} are denoted as $\mathcal{X}(i, :, :)$, $\mathcal{X}(:, i, :)$, and $\mathcal{X}(:, :, i)$ respectively. The (i, j, k) -th element is denoted as \mathcal{X}_{ijk} . For sim-

plicity, we also use $\mathbf{X}^{(i)}$ to denote the i -th frontal slice. $\mathcal{X}^{(i)} \in \mathbb{R}^{n_1 \times 1 \times n_3}$ denotes the tensor that only composed of the i -th lateral slice of \mathcal{X} . The inner product of tensors is denoted as $\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{ijk} \mathcal{X}_{ijk} \mathcal{Y}_{ijk}$. The Frobenius norm of tensor is denoted as $\|\mathcal{X}\|_F = \sqrt{\sum_{ijk} \mathcal{X}_{ijk}^2}$. We use $\text{fft}(\mathcal{X}, [1, 3]) = \overline{\mathcal{X}} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$ to denote performing DFT on all the tubes of $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$. The inverse FFT on $\overline{\mathcal{X}}$ can turn it back to the original tensor, i.e., $\mathcal{X} = \text{ifft}(\overline{\mathcal{X}}, [1, 3])$.

Definitions, Tensor Factorization and Tubal Rank

Unfold and Fold operators for a tensor are defined as

$$\text{Unfold}(\mathcal{X}) := [\mathbf{X}^{(1)}; \dots; \mathbf{X}^{(n_3)}] \quad (3)$$

$$\text{Fold}(\text{Unfold}(\mathcal{X})) := \mathcal{X}. \quad (4)$$

Denote the block circulant matrix of \mathcal{X} as

$$\text{bcirc}(\mathcal{X}) := \begin{bmatrix} \mathbf{X}^{(1)} & \mathbf{X}^{(n_3)} & \dots & \mathbf{X}^{(2)} \\ \mathbf{X}^{(2)} & \mathbf{X}^{(1)} & \dots & \mathbf{X}^{(3)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{X}^{(n_3)} & \mathbf{X}^{(n_3-1)} & \dots & \mathbf{X}^{(1)} \end{bmatrix}.$$

The above $\text{bcirc}(\mathcal{X})$ can be block diagonalized as

$$(\mathbf{F}_{n_3} \otimes \mathbf{I}_{n_1}) \cdot \text{bcirc}(\mathcal{X}) \cdot (\mathbf{F}_{n_3} \otimes \mathbf{I}_{n_2}) = \overline{\mathbf{X}}, \quad (5)$$

where \mathbf{F}_n denotes the n -dimensional discrete Fourier transformation matrix, \otimes denotes the Kronecker product and $\overline{\mathbf{X}}$ is defined as:

$$\overline{\mathbf{X}} := \text{bdiag}(\overline{\mathbf{X}}) := \text{diag}(\overline{\mathbf{X}}^{(1)}; \dots; \overline{\mathbf{X}}^{(n_3)}). \quad (6)$$

With the above definitions, we introduce the following arithmetic operations for tensors.

Definition 2 (Tensor-Tensor product (T-product)) (Kilmer and Martin 2011) The tensor product between tensors $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{Y} \in \mathbb{R}^{n_2 \times n_4 \times n_3}$ is defined as:

$$\mathcal{X} * \mathcal{Y} = \text{Fold}(\text{bcirc}(\mathcal{X}) \text{Unfold}(\mathcal{Y})) \in \mathbb{R}^{n_1 \times n_4 \times n_3}. \quad (7)$$

Definition 3 (Conjugate transpose) (Lu et al. 2020) The conjugate transpose of $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is \mathcal{X}^c that $\mathcal{X}^c(:, :, 1) = (\mathbf{X}^{(1)})^c$, $\mathcal{X}^c(:, :, n_3 + 2 - i) = (\mathbf{X}^{(i)})^c$ for $2 \leq i \leq n_3$, where \mathbf{X}^c is conjugate transpose of \mathbf{X} .

Definition 4 (Identity tensor) (Kilmer and Martin 2011) If $\mathcal{I}(:, :, 1) = \mathbf{I}_n$ and $\mathcal{I}(:, :, i) = \mathbf{0}_n$ for $2 \leq i \leq n_3$, then $\mathcal{I} \in \mathbb{R}^{n \times n \times n_3}$ is defined as the identity tensor.

Definition 5 (Orthogonal tensor) (Kilmer and Martin 2011) A tensor $\mathcal{Q} \in \mathbb{R}^{n \times n \times n_3}$ is defined as the orthogonal tensor if $\mathcal{Q} * \mathcal{Q}^c = \mathcal{Q}^c * \mathcal{Q} = \mathcal{I}$.

Definition 6 (Tensor inverse) (Kilmer and Martin 2011) An $n \times n \times n_3$ tensor \mathcal{X} has an inverse \mathcal{Y} if $\mathcal{X} * \mathcal{Y} = \mathcal{I}$ and $\mathcal{Y} * \mathcal{X} = \mathcal{I}$. If \mathcal{X} is invertible, we use \mathcal{X}^{-1} to denote its inverse.

Definition 7 (F-diagonal tensor) (Kilmer and Martin 2011) If all of frontal slices of \mathcal{X} are diagonal matrices, then \mathcal{X} is called an f -diagonal tensor.

Theorem 1 (t-SVD) (Lu et al. 2020) Let $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$. Then it can be factorized as

$$\mathcal{X} = \mathcal{U} * \mathcal{S} * \mathcal{V}^c, \quad (8)$$

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

Similar to matrices, the tensor QR factorization is defined as follows.

Theorem 2 (T-QR) (Kilmer and Martin 2011) Let $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$. Then it can be factorized as

$$\mathcal{X} = \mathcal{Q} * \mathcal{R}, \quad (9)$$

where $\mathcal{Q} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$ is orthogonal, and $\mathcal{R} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is an f -upper triangular tensor whose frontal slices are all upper triangular matrices.

Definition 8 (Tubal rank) (Lu et al. 2020) For a tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, its tubal rank is defined as the number of nonzero singular tubes of \mathcal{S} , where \mathcal{S} is the f -diagonal tensor obtained from t-SVD of \mathcal{X} . Specially,

$$\text{rank}_t(\mathcal{X}) = \#\{i : \mathcal{S}(i, i, :) \neq \mathbf{0}\}. \quad (10)$$

Lastly, we introduce the tensor spectral norm and condition number.

Definition 9 (Tensor spectral norm) (Lu et al. 2018) The spectral norm of $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is defined as

$$\|\mathcal{X}\| = \sigma_{\max}(\text{bcirc}(\mathcal{X})) = \max_{i \in n_3} \sigma_{\max}(\overline{\mathbf{X}}^{(i)}), \quad (11)$$

where $\sigma_{\max}(\mathbf{X})$ denotes maximum singular value of \mathbf{X} .

Definition 10 (Tensor condition number) The condition number of $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is defined as the condition number of $\text{bcirc}(\mathcal{X})$ as

$$\kappa(\mathcal{X}) = \kappa(\text{bcirc}(\mathcal{X})) = \frac{\sigma_{\max}(\text{bcirc}(\mathcal{X}))}{\sigma_{\min}(\text{bcirc}(\mathcal{X}))}, \quad (12)$$

where the $\sigma_{\min}(\text{bcirc}(\mathcal{X}))$ denotes the smallest nonzero singular value of $\text{bcirc}(\mathcal{X})$.

If the condition number $\kappa(\mathcal{X})$ is close to 1, the tensor \mathcal{X} is said to be well-conditioned. Conversely, if the condition number $\kappa(\mathcal{X})$ is large, then \mathcal{X} is deemed ill-conditioned (Tong et al. 2022a). From Figure 1 and Figure 2, recovering the ill-conditioned low-rank tensor \mathcal{X}^* in the TCS problems is more challenging.

Algorithms and Theoretical Results

Given the local TCS model (2), a natural formulation is to minimize the fitting loss

$$\hat{f}(\mathcal{X}) := \sum_{i=1}^{n_2} \sum_{j=1}^m (y_{ji} - \langle \mathcal{A}_i(:, j, :), \mathcal{X}(:, i, :) \rangle)^2. \quad (13)$$

under the constraint that the tubal rank of \mathcal{X} is at most equal to r . Based on the t-SVD, we can reparameterize the variable as $\mathcal{X} = \mathcal{U} * \mathcal{V}$ to incorporate the low-rank constraint, with $\mathcal{U} \in \mathbb{R}^{n_1 \times r \times n_3}$ satisfying the orthogonality constraint

$\mathcal{U}^c * \mathcal{U} = \mathcal{I}_r$ and $\mathcal{V} \in \mathbb{R}^{r \times n_2 \times n_3}$. Overall, the optimization problem is written as:

$$\begin{aligned} \min_{\mathcal{U}, \mathcal{V}} f(\mathcal{U}, \mathcal{V}) &= \sum_{i=1}^{n_2} \sum_{j=1}^m (y_{ji} - \langle \mathcal{A}_i(:, j, :), \mathcal{U} * \mathcal{V}(:, i, :) \rangle)^2 \\ \text{s.t.} \quad \mathcal{U}^c * \mathcal{U} &= \mathcal{I}_r. \end{aligned} \quad (14)$$

Note that the reparameterization also significantly reduces the number of variables under small r , unlocking the potential of designing low-complexity algorithms. However, as a tradeoff, it introduces nonconvexity through the tensor product in the objective and the orthogonality constraint.

This section proposes algorithms to solve (14) to the *global minimum* despite nonconvexity. The approach consists of two stages. The first stage employs a spectral initialization to find an initial point \mathcal{U}_0 that is sufficiently close to the minimizer, thus providing a warm start for the second stage. The second stage is based on a local search strategy that alternately optimizes \mathcal{U} and \mathcal{V} according to (14). In the remainder of this section, we provide a detailed introduction to the two stages.

The following mild assumptions are made on \mathcal{X}^* and the sensing operator $\{\mathcal{A}_i\}_{i=1}^{n_3}$ for obtaining our results.

Assumption 1 *The ground truth $\mathcal{X}^* \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ has tubal rank $r \ll \min\{n_1, n_2, n_3\}$. Its skinny t-SVD is $\mathcal{X}^* = \mathcal{U}^* * \mathcal{S}^* * (\mathcal{V}^*)^c$ that $\mathcal{U}^* \in \mathbb{R}^{n_1 \times r \times n_3}$, $\mathcal{S}^* \in \mathbb{R}^{r \times r \times n_3}$, $\mathcal{V}^* \in \mathbb{R}^{r \times n_2 \times n_3}$ and $\mathcal{Z}^* = \mathcal{S}^* * (\mathcal{V}^*)^c$. There exists a finite constant μ such that $\max_{i \in [n_2]} \|\mathcal{Z}^*(:, i, :)\|_F \leq \mu \sqrt{\frac{r}{n_2}} \|\mathcal{X}^*\|$.*

This assumption is similar to the tensor incoherence condition in the low tubal rank tensor recovery literature (Zhang and Aeron 2016; Lu et al. 2018, 2020; Zhang et al. 2020a) and ensures that our problem remains well-posed.

Assumption 2 *Each sensing tensor $\mathcal{A}_i \in \mathbb{R}^{n_1 \times m \times n_3}$, $\forall i \in [n_2]$, has i.i.d. standard Gaussian entries.*

Stage I: Truncated Spectral Initialization

The idea of spectral-based initialization, which is used for providing a “warm start” within the basin of attraction for \mathcal{X}^* , has been extensively utilized in various non-convex low-rank matrix and tensor recovery problems (Cai et al. 2019; Liu et al. 2024). Inspired by the truncation technique (Chen and Candes 2015; Wang, Giannakis, and Eldar 2017; Vaswani 2024), we design the following initialization method for the local sensing model (2). Specifically, let

$$\hat{\mathcal{X}}_0(:, i, :) = \frac{1}{m_0} \sum_{j=1}^{m_0} y_{ji} \mathcal{A}_i(:, j, :) \cdot \mathbf{1}_{\{|y_{ji}| \leq \sqrt{\alpha}\}}, \quad (15)$$

where $\mathbf{1}_{\{|y_{ji}| \leq \sqrt{\alpha}\}}$ denotes an indicator function that is equal to 1 when $|y_{ji}| \leq \sqrt{\alpha}$ and 0 otherwise. m_0 is the number of sensing matrix for per lateral slice. The α is the threshold in truncation and its formula is given in the subsequent theorem. The reason for truncation is that we can use tight sample complexity to bound the concentration of $\hat{\mathcal{X}}_0$ on \mathcal{X}^* . Subsequently, we perform the QR decomposition:

$$\hat{\mathcal{X}}_0 = \mathcal{Q}_0 * \mathcal{R}_0, \quad (16)$$

and initialize the orthogonal tensor $\mathcal{U}_0 \in \mathbb{R}^{n_1 \times r \times n_3}$ to be the first r lateral slices of \mathcal{Q}_0 , i.e.,

$$\mathcal{U}_0 = \mathcal{Q}_0(:, 1 : r, :).$$

The following measure defines the sine of the largest angle between tensor subspaces spanned by their lateral slices.

Definition 11 (Principal angle distance) *For two orthogonal tensors $\mathcal{A}_1, \mathcal{A}_2 \in \mathbb{R}^{n_1 \times r \times n_3}$, the principal angle distance between \mathcal{A}_1 and \mathcal{A}_2 is defined as*

$$\text{Dis}(\mathcal{A}_1, \mathcal{A}_2) = \|(\mathcal{I}_r - \mathcal{A}_1 * \mathcal{A}_1^c) * \mathcal{A}_2\|. \quad (17)$$

Based on this measure, we can prove the effectiveness of the proposed initialization method in the following theorem.

Theorem 3 *Consider the TCS model (2) under Assumption 1 and 2. The initialization \mathcal{U}_0 in Algorithm 1 satisfies*

$$\text{Dis}(\mathcal{U}_0, \mathcal{U}^*) \leq \frac{0.016}{\sqrt{r} \kappa^2} \quad (18)$$

with a probability at least

$$\begin{aligned} 1 - \exp\left(c_1(n_1 + n_2) \log n_3 - \frac{c_2 m_0 n_2}{\kappa^8 \mu^2 n_3 r^2}\right) \\ - \exp\left(-c_3 \frac{m_0 n_2}{\kappa^8 \mu^2 r^2}\right), \end{aligned} \quad (19)$$

where κ is the tensor condition number of \mathcal{X}^* and c_1, c_2, c_3 are universal constants that are independent of model parameters.

Theorem 3 immediately implies the following sample complexity for our spectral initialization scheme.

Corollary 1 *In the same setting as Theorem 3, if the sample size m_0 for initialization satisfies*

$$m_0 n_2 \gtrsim \kappa^8 \mu^2 r^2 n_3 (n_1 + n_2) \log n_3, \quad (20)$$

then (18) holds with probability at least $1 - \frac{1}{(n_1 + n_3)^{10}}$.

Compared to the total number of entries in \mathcal{U}^* and \mathcal{Z}^* , which is $r n_3 (n_1 + n_2)$, Corollary 1 shows a good initialization can be achieved with a sample size only having an additional factor of $r \log n_3$ (modulus constants κ, μ).

Stage II: Local Search

The second stage concerns iteratively refining the initial point \mathcal{U}_0 computed by Stage I by local search. Since the objective f in (14) is bi-convex in \mathcal{U} and \mathcal{V} , based on this structure, we first propose the Alt-PGD-Min algorithm that alternately updates these two factors.

The Alt-PGD-Min Algorithm. Let \mathcal{U}_t and \mathcal{V}_t be the values of \mathcal{U} and \mathcal{V} at iteration t .

1) *Exact minimization for \mathcal{V}* : Fixing \mathcal{U}_t , the lateral slices of \mathcal{V} are decoupled in problem (14). Thus, we can update each lateral slice $\mathcal{V}(:, i, :)$ in parallel by solving the following minimization problem

$$\mathcal{V}_t(i) \in \arg \min_{\mathcal{B} \in \mathbb{R}^{r \times 1 \times n_3}} \sum_{j=1}^{m_c} (y_{ji} - \langle \mathcal{U}_t^c * \mathcal{A}_i(j), \mathcal{B} \rangle)^2, \quad (21)$$

which can be reformulated as follows based on Definition 2

$$\mathbf{v}_{t,i} \in \arg \min_{\mathbf{v} \in \mathbb{R}^{rn_3}} \|(\text{bcirc}(\mathbf{U}_t^c) \cdot \text{Unfold}(\mathcal{A}_i))^c \cdot \mathbf{v} - \mathbf{y}_i\|^2. \quad (22)$$

The problem (22) is a least squares problem. Thus, for every $i \in [n_2]$, a closed-form solution can be derived as follows:

$$\begin{aligned} \mathbf{H}_{t,i} &= \text{bcirc}(\mathbf{U}_t^c) \cdot \text{Unfold}(\mathcal{A}_i), \\ \mathbf{v}_{t,i} &= (\mathbf{H}_{t,i} \mathbf{H}_{t,i}^c)^{-1} \mathbf{H}_{t,i} \mathbf{y}_i, \\ \mathbf{V}_t(i) &= \text{Fold}(\mathbf{v}_{t,i}). \end{aligned} \quad (23)$$

2) *Projected gradient descent for \mathbf{U}* : Although for fixed \mathbf{V}_t , f is also convex in \mathbf{U} , in pursuit of computation-efficient update, instead of performing exact minimization, we employ a first-order gradient descent step to update \mathbf{U} (Gu et al. 2024), followed by a projection step onto the orthogonality constraint set. Specifically, we first compute

$$\begin{aligned} \hat{\mathbf{U}}_{t+1} &= \mathbf{U}_t - \eta \sum_{i=1}^{n_2} \sum_{j=1}^{m_c} (y_{ji} - \langle \mathbf{U}_t^c * \mathcal{A}_i(j), \mathbf{V}_t(i) \rangle) \\ &\quad \cdot \mathcal{A}_i(j) * (\mathbf{V}_t(i))^c, \end{aligned} \quad (24)$$

with step size $\eta > 0$. Then we obtain a tensor $\hat{\mathbf{Q}}_{t+1}$ by the QR decomposition $\hat{\mathbf{U}}_{t+1} = \hat{\mathbf{Q}}_{t+1} * \hat{\mathbf{R}}_{t+1}$. The updated of \mathbf{U} is given by

$$\mathbf{U}_{t+1} = \hat{\mathbf{Q}}_{t+1}(:, 1:r, :). \quad (25)$$

The complete Alt-PGD-Min algorithm is described in Algorithm 1. It is worth mentioning that we use the sample-splitting technique in Algorithm 1, where we divide the total samples and measurements pairs of each slice into $2T + 1$ groups as $\{\mathcal{A}_i^{(k)}\}_{k=1}^{2T+1}$. The last two groups are used for initialization that the sample size for each lateral slice is m_0 , and Alt-PGD-Min draws two fresh groups of samples from the remaining groups per iteration, where the sample size for each lateral slice is m_c . The splitting strategy ensures statistical independence of the samples across iterations, which is a key component to simplifying the convergence analysis and has been used in various low matrix and tensor learning algorithms (Hardt and Wootters 2014; Jain and Netrapalli 2015; Ding and Chen 2020; Cai, Li, and Xia 2022).

The computational complexity per iteration in Alt-PGD-Min for updating \mathbf{U} and \mathbf{V} is $\mathcal{O}(n_1 n_3 m_c r + n_3 m_c r + n_1 n_2 n_3 r + n_1 n_3 r^2)$ and $\mathcal{O}(n_1 n_2 n_3 m_c r + m_c (r n_3)^2 n_2 + (n_3 r)^3 n_2)$, respectively.

Theorem 4 *In the same setting as Theorem 3, if the initialization \mathbf{U}_0 satisfies (18) and $\eta = \frac{c_\eta}{m_c \|\mathcal{X}^*\|^2}$ with $c_\eta \leq 0.9$, then the iterates generated by Alt-PGD-Min satisfies*

$$\begin{aligned} \text{Dis}(\mathbf{U}_t, \mathbf{U}^*) &\leq \left(1 - \frac{0.84c_\eta}{\kappa^2}\right)^t \cdot \text{Dis}(\mathbf{U}_0, \mathbf{U}^*), \\ \|\mathcal{X}_t(i) - \mathcal{X}^*(i)\|_F &\leq 1.4 \text{Dis}(\mathbf{U}_t, \mathbf{U}^*) \cdot \|\mathcal{X}^*(i)\|_F \end{aligned} \quad (26)$$

for $\forall t \geq 0, i \in [n_2]$ with a probability at least

$$\begin{aligned} 1 - \exp\left(c_4(n_1 + r) \log n_3 - \frac{c_5 m_c n_2}{\kappa^4 \mu^2 n_3 r}\right) \\ - \exp(\log n_2 + r \log n_3 - c_6 m_c), \end{aligned} \quad (27)$$

Algorithm 1: Alt-PGD-Min/Alt-ScalePGD-Min

Input: Number of iteration T , total sensing tensors with sample splitting $\{\{\mathcal{A}_i^{(k)}\}_{k=1}^{2T+1}\}_{i=1}^{n_2}$, corresponding sample-splitting local measurements $\{\{\mathbf{y}_i^{(k)}\}_{k=1}^{2T+1}\}_{i=1}^{n_2}$, r, κ, μ, n_2 , step size η .

```

1: for  $t = 0, 1, \dots, T - 1$  do
2:    $\triangleright$  Update  $\mathbf{U}$ 
3:   if  $t = 0$  then  $\triangleright$  Initialization
4:     Set  $\mathcal{A}_i = \mathcal{A}_i^{(2T)}, \mathbf{y}_i = \mathbf{y}_i^{(2T)}, \forall i \in [n_2]$ ,
5:     Calculate  $\alpha = C^{\kappa^2 \mu^2} \frac{\sum_{i=1}^{n_2} \sum_{j=1}^{m_c} y_{ji}}{m n_2}$ ,
6:     Set  $\mathcal{A}_i = \mathcal{A}_i^{(2T+1)}, \mathbf{y}_i = \mathbf{y}_i^{(2T+1)}, \forall i \in [n_2]$ ,
7:     Construct  $\hat{\mathcal{X}}_0$  as (15),
8:     Conduct QR decomposition  $\hat{\mathcal{X}}_0 = \hat{\mathbf{Q}}_0 * \hat{\mathbf{R}}_0$ ,
9:     Initialize  $\mathbf{U}_0$  by top- $r$  lateral slices of  $\hat{\mathbf{Q}}_0$ .
10:  else
11:    for  $i = 1, 2, \dots, n_2$  do
12:       $\mathcal{A}_i = \mathcal{A}_i^{(T+t)}, \mathbf{y}_i = \mathbf{y}_i^{(T+t)}, \forall i \in [n_2]$ ,
13:       $\mathbf{H}_{t-1,i} = \text{bcirc}(\mathbf{U}_{t-1}^c) \cdot \text{Unfold}(\mathcal{A}_i)$ ,
14:       $\mathbf{b}_{t-1,i} = \mathbf{H}_{t-1,i}^c \text{Unfold}(\mathbf{V}_{t-1}(i)) - \mathbf{y}_i$ ,
15:       $\mathcal{T}_{t-1}(:, i, :) = \sum_{j=1}^m (\mathbf{b}_{t-1,i})_j \mathcal{A}_i(:, j, :)$ ,
16:    end for
17:     $\hat{\mathbf{U}}_t = \mathbf{U}_t - \eta \mathcal{T}_{t-1} * \mathbf{V}_{t-1}^c$ ,
18:     $\hat{\mathbf{U}}_t = \mathbf{U}_t - \eta \mathcal{T}_{t-1} * \mathbf{V}_{t-1}^c * (\mathbf{V}_{t-1} * \mathbf{V}_{t-1}^c)^{-1}$ ,
19:    Calculate  $\mathbf{U}_t$  as (25) by QR decomposition,
20:  end if
21:   $\triangleright$  Update  $\mathbf{V}$ 
22:  for  $i = 1, 2, \dots, n_2$  do
23:     $\mathcal{A}_i = \mathcal{A}_i^{(t+1)}, \mathbf{y}_i = \mathbf{y}_i^{(t+1)}, \forall i \in [n_2]$ ,
24:     $\mathbf{H}_{t,i} = \text{bcirc}(\mathbf{U}_t^c) \cdot \text{Unfold}(\mathcal{A}_i)$ ,
25:     $\mathbf{v}_{t,i} = (\mathbf{H}_{t,i} \mathbf{H}_{t,i}^c)^{-1} \mathbf{H}_{t,i} \mathbf{y}_i$ ,
26:     $\mathbf{V}_t(i) = \text{Fold}(\mathbf{v}_{t,i})$ .
27:  end for
28:   $\triangleright$  Update  $\mathcal{X}$ 
29:   $\mathcal{X}_t = \mathbf{U}_t * \mathbf{V}_t$ ,
30: end for

```

Output: Recover tensor \mathcal{X}_{T-1} .

where c_4, c_5, c_6 are universal positive constants independent from model parameters.

Theorem 4 shows that even (14) is non-convex, if the initialization \mathbf{U}_0 is sufficiently close to the minimizer, the iterations of Alt-PGD-Min will converge at a linear rate to \mathcal{X}^* .

Corollary 2 *In the same setting as Theorem 4, if the $\eta = \frac{0.8}{m_c \|\mathcal{X}^*\|}$ and the sample size m_c satisfies*

$$m_c \gtrsim \max\{\kappa^4 \mu^2 r n_1 n_3 \log n_3 / n_2, \log n_2, r \log n_3\} \quad (28)$$

then it takes $T = c_7 \kappa^2 \log \frac{1}{\epsilon}$ iterations for Alt-PGD-Min to achieve ϵ -accuracy recovery, i.e.,

$$\begin{aligned} \text{Dis}(\mathbf{U}_T, \mathbf{U}^*) &\leq \epsilon, \\ \|\mathcal{X}_T(i) - \mathcal{X}^*(i)\|_F &\leq 1.4\epsilon \|\mathcal{X}^*(i)\|_F, \forall i \in [n_2] \end{aligned} \quad (29)$$

with probability at least $1 - \frac{1}{(n_1+r)^{10}}$.

The sample complexity given by (28) scales linearly with r , showing improved dependence on r compared to that for the initialization Stage I. When the number of lateral slices is large enough such that $n_2 \gtrsim \kappa^4 \mu^2 n_1 \log n_3$, the order of m_c becomes $\mathcal{O}(rn_3)$, which is significantly smaller than the size of lateral slice $n_1 n_3$ as $r \ll \min(n_1, n_2)$.

Combining the results of Corollary 1 and Corollary 2, we can immediately conclude the overall sample complexity of Alt-PGD-Min as follows.

Corollary 3 Consider the TCS model (2) under Assumption 1 and 2. For Alt-PGD-Min to achieve ϵ -accuracy recovery as described by (29) with high probability at least $1 - \frac{2}{(n_1+r)^{10}}$, the total sample complexity m for each lateral slice is

$$mn_2 \gtrsim \kappa^6 \mu^2 rn_3 \log n_3 \left(\kappa^2 r (n_1 + n_2) + n_1 \log \frac{1}{\epsilon} \right) \quad (30)$$

and $m \gtrsim \kappa^2 \max \{ \log n_2, r \log n_3 \} \log \frac{1}{\epsilon}$.

The total sample complexity in (30) comprises two parts: one is from initialization and the other is from iterative refinements. The dependency of the sample complexity on recovery accuracy ϵ is because of the sample splitting strategy introduced in the algorithm, which is common in all the analyses where such a technique is adopted (Jain, Netrapalli, and Sanghavi 2013; Hardt and Wootters 2014; Ding and Chen 2020; Vaswani 2024).

Alt-ScalePGD-Min: Acceleration by Preconditioning.

To mitigate the influence of large κ and improve the algorithm efficiency, especially for ill-conditioned problems, we propose to accelerate by pre-conditioning the gradient step that is sensitive to κ . Recall (24) and let $\mathbf{b}_{t,i} := \mathbf{H}_{t,i}^c \mathbf{v}_{t,i} - \mathbf{y}_i, i \in [n_2]$ and $\mathcal{T}_t(\cdot, i, \cdot) := \sum_{j=1}^{m_c} (\mathbf{b}_{t,i})_j \cdot \mathcal{A}_i(\cdot, j, \cdot)$, we rewrite the updating step as

$$\hat{\mathbf{u}}_{t+1} = \mathbf{u}_t - \eta \mathcal{T}_t * \mathbf{v}_t^c. \quad (31)$$

Comparing to the gradient step (31), the key difference of Alt-ScalePGD-Min is that it preconditions the search direction of \mathbf{u}_t by inverse of $\mathbf{v}_t * \mathbf{v}_t^c$, i.e.,

$$\hat{\mathbf{u}}_{t+1} = \mathbf{u}_t - \eta \mathcal{T}_t * \mathbf{v}_t^c * (\mathbf{v}_t * \mathbf{v}_t^c)^{-1}. \quad (32)$$

Note that the inverse in (32) is easy to compute because the related tensor has a size of $r \times r \times n_3$, significantly smaller than the dimension of the tensor factors. Thus, each iteration of scaled GD incurs minor additional complexity $\mathcal{O}(n_1 r^2 n_3 + r^3 n_3)$ than the GD of Alt-PGD-Min in (31).

The convergence of Alt-ScalePGD-Min is given in the following theorem.

Theorem 5 Consider the TCS model (2) under Assumption 1 and 2. Let the step size $\eta = \frac{c_n}{m_c}$ and all other parameters are set the same as Theorem 4, then the iterates generated by Alt-ScalePGD-Min satisfy

$$\text{Dis}(\mathbf{u}_t, \mathbf{u}^*) \leq (1 - 0.89c_\eta)^t \cdot \text{Dis}(\mathbf{u}_0, \mathbf{u}^*) \quad (33)$$

with at least the same probability as (27).

Corollary 4 In the same setting as Theorem 5, if $\eta = \frac{0.8}{m_c}$ and sample size for each lateral slice m_c satisfies (28), then to obtain the ϵ -accuracy recovery as described by (29), the iteration complexity for Alt-Scale-GD is

$$T = c_7 \log \frac{1}{\epsilon}. \quad (34)$$

Theorem 5 and Corollary 4 show Alt-ScalePGD-Min converges linearly at a rate that is independent of the condition number κ , significantly improving over the $\mathcal{O}(\kappa^2 \log \frac{1}{\epsilon})$ complexity of Alt-PGD-Min.

Corollary 5 Consider the TCS model (2) under Assumption 1 and 2. The total sample complexity m for each lateral slice to achieve ϵ -accuracy recovery as (29) with high probability at least $1 - \frac{2}{(n_1+r)^{10}}$ is

$$mn_2 \gtrsim \kappa^4 \mu^2 rn_3 \log n_3 \left(\kappa^4 r (n_1 + n_2) + n_1 \log \frac{1}{\epsilon} \right) \quad (35)$$

and $m \gtrsim \max \{ \log n_2, r \log n_3 \} \log \frac{1}{\epsilon}$.

Due to the same initialization, the first term in (30) and (35) are the same. However, Alt-ScalePGD-Min improves over Alt-PGD-Min by a factor of κ^2 in the second term due to improved convergence rate. When the recovery accuracy is sufficiently high such that the second term dominates the first, i.e., $\log \frac{1}{\epsilon} \gtrsim \kappa^4 r (1 + \frac{n_2}{n_1})$, the total sample complexity of Alt-ScalePGD-Min is $\mathcal{O}(\kappa^4 \mu^2 rn_1 n_3 \log n_3 \log \frac{1}{\epsilon})$, significantly improving upon the $\mathcal{O}(\kappa^6 \mu^2 rn_1 n_3 \log n_3 \log \frac{1}{\epsilon})$ of Alt-PGD-Min for large κ .

Experiments

We evaluate our proposed methods on both synthetic and real-world data. Since model (2) has not been studied in existing works, we can only compare our method with the low-rank matrix column-wise CS method (LRcCS) (Nayer and Vaswani 2022), which is closest to our TCS model in application. To compare with this method, we conduct Unfold(\mathcal{X}^*) operation which reshapes each lateral slice into a vector. The performance of algorithms is measured by the relative recovery error $\frac{\|\mathcal{X}_t - \mathcal{X}^*\|_F}{\|\mathcal{X}^*\|_F}$, which is plotted concerning the iteration number.

Synthetic Data

We generate a synthetic tensor with $n_1 = n_3 = 20, n_2 = 400, r = 4$, of which the details are in the Appendix. The sample sizes are $m_0 = 200$ and $m_c = 100$ without sample splitting. We test three algorithms with the same step size and sample sizes under different $\kappa = 1, 2, 4$. The results are plotted in Figure 1, which shows that both our proposed methods have linear convergence rates while LRcCS fails in all cases. The convergence rate of Alt-PGD-Min becomes slower with increasing κ while Alt-ScalePGD-Min converges with independence on κ , with all curves overlapping.

In the second setting, the data generation is the same as the first. However, we validate the performance under

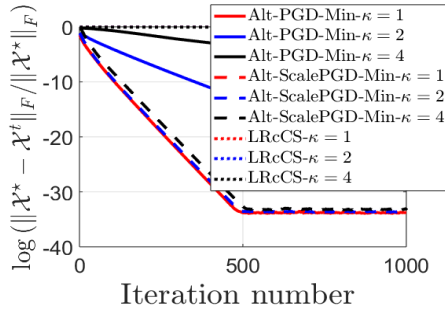


Figure 1: Spectral initialization with $m_0 = 200, m_c = 100$.

random initialization, which has i.i.d. standard Gaussian entries. Without good initialization, we increase the m_c slightly to $m_c = 120$. The results depicted in Figure 2 show that both of the proposed algorithms converge with linear rates after a small number of iterations during the initial phase while LRcCS still does not work. With larger κ , Alt-PGD-Min slows down significantly while the convergence speed of Alt-ScalePGD-Min remains almost the same with almost negligible initial phases.

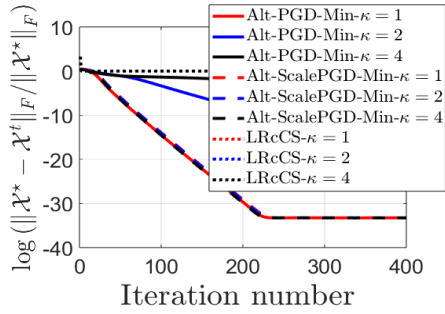


Figure 2: Random initialization with $m_c = 120$.

Video Compressed Sensing

We test the proposed TCS model (2) in the plane video sequence (only approximately low tubal rank) that has been used in previous work such as (Nayer, Narayanamurthy, and Vaswani 2019). It has 105 frames and each frame has been resized into 48×64 . We use the same samples with size $m = 1600$ for initialization and iteration. We set $r = 10$ for three methods. For each method, we tune the step size to guarantee it converges and achieves performance as good as possible. The visual and quantitative comparison results are shown in Figure 3 and Figure 4, respectively. Alt-ScalePGD-Min has the fastest convergence rate with the best recovery performance (Figure 3. (a) selected at 20-th iteration). Both Alt-PGD-Min and LRcCS converge very slowly with unsatisfied performance (Figure 3. (c) and (d) are selected by running 8000 iterations). This is because the video has very large matrix and tensor condition numbers that result in slow convergence rates.

We can observe that while Alt-PGD-Min outperforms LRcCS in synthetic data experiments, it performs worse in

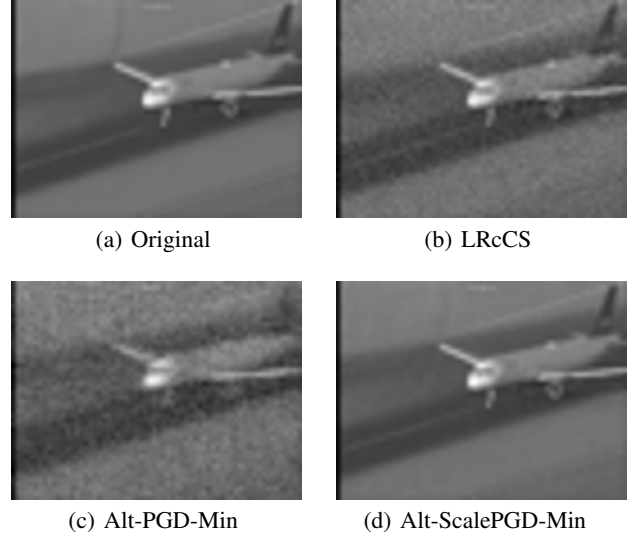


Figure 3: Visualization of frame-7 in recovered videos.

video compressive sensing. This discrepancy is because the synthetic data is generated to be exactly low-tubal-rank with a controlled tensor condition number. In contrast, the video data is only approximately low tubal rank and has a significantly larger tensor condition number compared to its reshaped matrix condition number. As a result, given the same number of iterations, Alt-PGD-Min performs worse than LRcCS in the video CS scenario.

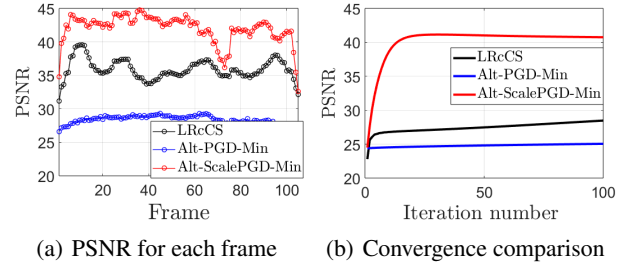


Figure 4: Quantitative comparison

Conclusion

We introduced a novel local TCS model for low-tubal-rank tensors and developed two algorithms to solve this model with theoretical guarantees. The numerical results demonstrated the effectiveness of our method. There are several problems worth studying in the future. For instance, it would be valuable to investigate whether the current theoretical guarantees can be established without the sample-splitting technique. Additionally, exploring the possibility of achieving global convergence under random initialization, without spectral initialization, remains challenging. Lastly, improving the complexity dependence on the tensor condition number is another interesting direction for future work.

Acknowledgments

The work of Fan was partially supported by the Youth Program 62106211 of the National Natural Science Foundation of China.

References

- Ahmed, T.; Raja, H.; and Bajwa, W. U. 2020. Tensor regression using low-rank and sparse Tucker decompositions. *SIAM Journal on Mathematics of Data Science*, 2(4): 944–966.
- Cai, C.; Li, G.; Poor, H. V.; and Chen, Y. 2019. Nonconvex low-rank tensor completion from noisy data. *Advances in neural information processing systems*, 32.
- Cai, J.-F.; Li, J.; and Xia, D. 2022. Provable tensor-train format tensor completion by riemannian optimization. *Journal of Machine Learning Research*, 23(123): 1–77.
- Candès, E. J.; Romberg, J.; and Tao, T. 2006. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on information theory*, 52(2): 489–509.
- Chen, H.; Raskutti, G.; and Yuan, M. 2019. Non-convex projected gradient descent for generalized low-rank tensor regression. *Journal of Machine Learning Research*, 20(5): 1–37.
- Chen, Y.; and Candes, E. 2015. Solving random quadratic systems of equations is nearly as easy as solving linear systems. *Advances in Neural Information Processing Systems*, 28.
- Ding, L.; and Chen, Y. 2020. Leave-one-out approach for matrix completion: Primal and dual analysis. *IEEE Transactions on Information Theory*, 66(11): 7274–7301.
- Fan, J. 2022. Multi-Mode Deep Matrix and Tensor Factorization. In *International Conference on Learning Representations*.
- Fan, J.; Ding, L.; Yang, C.; Zhang, Z.; and Udell, M. 2023. Euclidean-Norm-Induced Schatten-p Quasi-Norm Regularization for Low-Rank Tensor Completion and Tensor Robust Principal Component Analysis. *Transactions on Machine Learning Research*.
- Gong, M.; and Zhang, G. 2024. Non-local tensor sparse representation and tensor low rank regularization for dynamic MRI reconstruction. *International Journal of Machine Learning and Cybernetics*, 15(2): 493–503.
- Gu, Y.; Song, Z.; Yin, J.; and Zhang, L. 2024. Low Rank Matrix Completion via Robust Alternating Minimization in Nearly Linear Time. In *The Twelfth International Conference on Learning Representations*.
- Han, R.; Willett, R.; and Zhang, A. R. 2022. An optimal statistical and computational framework for generalized tensor estimation. *The Annals of Statistics*, 50(1): 1–29.
- Hardt, M.; and Wootters, M. 2014. Fast matrix completion without the condition number. In *Conference on learning theory*, 638–678. PMLR.
- Hou, J.; Zhang, F.; Qiu, H.; Wang, J.; Wang, Y.; and Meng, D. 2021. Robust low-tubal-rank tensor recovery from binary measurements. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8): 4355–4373.
- Jain, P.; and Netrapalli, P. 2015. Fast exact matrix completion with finite samples. In *Conference on Learning Theory*, 1007–1034. PMLR.
- Jain, P.; Netrapalli, P.; and Sanghavi, S. 2013. Low-rank matrix completion using alternating minimization. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, 665–674.
- Kilmer, M. E.; and Martin, C. D. 2011. Factorization strategies for third-order tensors. *Linear Algebra and its Applications*, 435(3): 641–658.
- Kolda, T. G.; and Bader, B. W. 2009. Tensor decompositions and applications. *SIAM review*, 51(3): 455–500.
- Kuzmin, S.; Mikhailova, V.; Dyakonov, I.; and Straupe, S. 2024. Learning the tensor network model of a quantum state using a few single-qubit measurements. *Physical Review A*, 109(5): 052616.
- Lee, K.; Sharma, R. S.; Junge, M.; and Romberg, J. 2023. Approximately low-rank recovery from noisy and local measurements by convex program. *Information and Inference: A Journal of the IMA*, 12(3): 1612–1654.
- Li, X.; Ye, Y.; and Xu, X. 2017. Low-rank tensor completion with total variation for visual data inpainting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.
- Li, Z.; Wang, Y.; Zhao, Q.; Zhang, S.; and Meng, D. 2022. A tensor-based online RPCA model for compressive background subtraction. *IEEE Transactions on Neural Networks and Learning Systems*, 34(12): 10668–10682.
- Liu, X.; Hou, J.; Peng, J.; Wang, H.; Meng, D.; and Wang, J. 2023a. Tensor compressive sensing fused low-rankness and local-smoothness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 8879–8887.
- Liu, X.-Y.; Aeron, S.; Aggarwal, V.; and Wang, X. 2019. Low-tubal-rank tensor completion using alternating minimization. *IEEE Transactions on Information Theory*, 66(3): 1714–1737.
- Liu, X.-Y.; Huang, Q.; Han, X.; Wu, B.; Kong, L.; Walid, A.; and Wang, X. 2023b. Real-Time Decoding of Snapshot Compressive Imaging Using Tensor FISTA-Net. *IEEE Transactions on Neural Networks and Learning Systems*.
- Liu, Z.; Han, Z.; Tang, Y.; Zhao, X.-L.; and Wang, Y. 2024. Low-Tubal-Rank Tensor Recovery via Factorized Gradient Descent. *arXiv preprint arXiv:2401.11940*.
- Lu, C.; Feng, J.; Chen, Y.; Liu, W.; Lin, Z.; and Yan, S. 2020. Tensor Robust Principal Component Analysis with A New Tensor Nuclear Norm. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.
- Lu, C.; Feng, J.; Lin, Z.; and Yan, S. 2018. Exact low tubal rank tensor recovery from Gaussian measurements. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2504–2510.
- Luo, Y.; and Zhang, A. R. 2023. Low-rank tensor estimation via riemannian gauss-newton: Statistical optimality and

- second-order convergence. *The Journal of Machine Learning Research*, 24(1): 18274–18321.
- Ma, J.; Liu, X.-Y.; Shou, Z.; and Yuan, X. 2019. Deep tensor admm-net for snapshot compressive imaging. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10223–10232.
- Moothedath, S.; and Vaswani, N. 2024. Decentralized Low Rank Matrix Recovery from Column-Wise Projections by Alternating GD and Minimization. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 12936–12940. IEEE.
- Mu, C.; Huang, B.; Wright, J.; and Goldfarb, D. 2014. Square deal: Lower bounds and improved relaxations for tensor recovery. In *International conference on machine learning*, 73–81. PMLR.
- Nayer, S.; Narayanamurthy, P.; and Vaswani, N. 2019. Phaseless PCA: Low-rank matrix recovery from column-wise phaseless measurements. In *International Conference on Machine Learning*, 4762–4770. PMLR.
- Nayer, S.; and Vaswani, N. 2022. Fast and sample-efficient federated low rank matrix recovery from column-wise linear and quadratic projections. *IEEE Transactions on Information Theory*, 69(2): 1177–1202.
- Ran, S.-J.; Sun, Z.-Z.; Fei, S.-M.; Su, G.; and Lewenstein, M. 2020. Tensor network compressed sensing with unsupervised machine learning. *Physical Review Research*, 2(3): 033293.
- Rauhut, H.; Schneider, R.; and Stojanac, Ž. 2017. Low rank tensor recovery via iterative hard thresholding. *Linear Algebra and its Applications*, 523: 220–262.
- Recht, B.; Fazel, M.; and Parrilo, P. A. 2010. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM review*, 52(3): 471–501.
- Shi, Z.; Han, J.; Zheng, T.; and Li, J. 2013. Guarantees of augmented trace norm models in tensor recovery. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, 1670–1676.
- Singh, A. P.; and Vaswani, N. 2024. Byzantine Resilient and Fast Federated Few-Shot Learning. In *Forty-first International Conference on Machine Learning*.
- Srinivasa, R. S.; Kim, S.; and Lee, K. 2023. Sketching low-rank matrices with a shared column space by convex programming. *IEEE Journal on Selected Areas in Information Theory*, 4: 54–60.
- Srinivasa, R. S.; Lee, K.; Junge, M.; and Romberg, J. 2019. Decentralized sketching of low rank matrices. *Advances in Neural Information Processing Systems*, 32.
- Tong, T.; Ma, C.; Prater-Bennette, A.; Tripp, E.; and Chi, Y. 2022a. Scaling and scalability: Provable nonconvex low-rank tensor completion. In *International Conference on Artificial Intelligence and Statistics*, 2607–2617. PMLR.
- Tong, T.; Ma, C.; Prater-Bennette, A.; Tripp, E.; and Chi, Y. 2022b. Scaling and scalability: Provable nonconvex low-rank tensor estimation from incomplete measurements. *Journal of Machine Learning Research*, 23(163): 1–77.
- Tucker, L. R. 1966. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3): 279–311.
- Vaswani, N. 2024. Efficient federated low rank matrix recovery via alternating gd and minimization: A simple proof. *IEEE Transactions on Information Theory*.
- Wang, A.; Li, C.; Jin, Z.; and Zhao, Q. 2020. Robust Tensor Decomposition via Orientation Invariant Tubal Nuclear Norms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 6102–6109.
- Wang, G.; Giannakis, G. B.; and Eldar, Y. C. 2017. Solving systems of random quadratic equations via truncated amplitude flow. *IEEE Transactions on Information Theory*, 64(2): 773–794.
- Wu, T.; and Fan, J. 2024. Smooth Tensor Product for Tensor Completion. *IEEE Transactions on Image Processing*.
- Wu, T.; Gao, B.; Fan, J.; and Xue, J. 2022. Low-rank tensor completion based on self-adaptive learnable transforms. *IEEE Transactions on Neural Networks and Learning Systems*.
- Wu, T.; and Sun, Y. 2024. Implicit Regularization of Decentralized Gradient Descent for Sparse Regression. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- Yu, Y.; Jin, J.; Liu, F.; and Crozier, S. 2014. Multidimensional compressed sensing MRI using tensor decomposition-based sparsifying transform. *PloS one*, 9(6): e98441.
- Zhang, F.; Wang, J.; Wang, W.; and Xu, C. 2020a. Low-tubal-rank plus sparse tensor recovery with prior subspace information. *IEEE transactions on pattern analysis and machine intelligence*, 43(10): 3492–3507.
- Zhang, F.; Wang, W.; Huang, J.; Wang, J.; and Wang, Y. 2020b. RIP-based performance guarantee for low-tubal-rank tensor recovery. *Journal of Computational and Applied Mathematics*, 374: 112767.
- Zhang, Z.; and Aeron, S. 2016. Exact tensor completion using t-SVD. *IEEE Transactions on Signal Processing*, 65(6): 1511–1526.
- Zhang, Z.; Ely, G.; Aeron, S.; Hao, N.; and Kilmer, M. 2014. 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*, 3842–3849.
- Zhou, P.; Lu, C.; Lin, Z.; and Zhang, C. 2017. Tensor factorization for low-rank tensor completion. *IEEE Transactions on Image Processing*, 27(3): 1152–1163.