ADVANCES IN MATRIX AND TENSOR ANALYSIS: FLEXIBLE AND ROBUST SAMPLING MODELS, ALGORITHMS, AND APPLICATIONS

By

Bowen Su

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Applied Mathematics—Doctor of Philosophy

2025

ABSTRACT

This thesis investigates robust and flexible methods for matrix and tensor analysis, which are fundamental in data science. The primary focus of this work is the development of Guaranteed Sampling Flexibility for Low-Tubal-Rank Tensor Completion, a project aimed at addressing the limitations of existing sampling methods for tensor completion, such as Bernoulli and t-CUR sampling, which often lack flexibility across diverse applications.

To overcome these challenges, we introduce Tensor Cross-Concentrated Sampling (t-CCS), an extension of the matrix Cross-Concentrated Sampling (CCS) model to tensors, and propose a novel non-convex algorithm, Iterative Tensor CUR Completion (ITCURC), specifically tailored for t-CCS-based tensor completion. Theoretical analysis provides sufficient conditions for low-rank tensor recovery and presents a detailed sampling complexity analysis. These findings are further validated through extensive testing on both real-world and synthetic datasets.

In addition to the main project, this thesis includes another one complementary study. The study explores the robustness of CCS model for matrix completion, a recent approach demonstrated to effectively capture cross-concentrated data dependencies. However, its robustness to sparse outliers has remained underexplored. To address this gap, we propose the Robust CCS Completion problem and develop a non-convex iterative algorithm, Robust CUR Completion (RCURC). Empirical results on synthetic and real-world datasets demonstrate that RCURC is both efficient and robust against outliers, making it a powerful tool for recovering incomplete data.

Collectively, these projects advance the robustness and flexibility of matrix and tensor methods, enhancing their applicability in complex, real-world data environments.

ACKNOWLEDGEMENTS

As I reach the culmination of my Ph.D. journey, I wish to extend my deepest gratitude to the Mathematics Department and the following professors whose support and guidance have been instrumental to my growth.

I would first like to express my sincere gratitude to my committee chair and advisor, Professor Andrew Christlieb. His expertise in Computational Science and Engineering has been a constant source of inspiration, significantly shaping my research perspective and deepening my understanding of high-performance computing. His invaluable advice and steadfast encouragement, particularly during the most challenging moments of my graduate studies, have left a lasting impact on both my personal and professional development. I would like long-term opportunities to work with Prof. Andrew on the development of low-rank tensor approximation techniques for high-dimensional PDEs. I am sincerely thankful to Professor Ekaterina Rapinchuk for her unwavering support and academic guidance. Her encouragement and insightful advice during my most difficult times provided me with strength and motivation. Her commitment to my success, from serving on my Ph.D. committee from the very beginning to offering steadfast support throughout my journey, has been truly indispensable. I would like long-term opportunities to collaborate with Prof. Ekaterina on the machine learning projects. I would like to express my sincere gratitude to Professor Yuying Xie for his continuous support and guidance. His encouragement has been invaluable, helping me stay focused and motivated throughout my doctoral journey. I would like long-term opportunities to collaborate with Prof. Xie on the AI for science projects. My heartfelt appreciation extends to Professor Mark Iwen for his generosity in sharing his expertise and professional wisdom. His thoughtful guidance and enduring support during his service on my Ph.D. committee have been invaluable.

Being part of the Department of Mathematics has been both an honor and a transformative experience. I am deeply grateful for the knowledge, skills, and unwavering support I have received. I sincerely extend my heartfelt appreciation to Professor Jeffery and Prof Rajesh for their dedication to fostering a supportive, friendly and enriching academic environment. Additionally, I am

deeply grateful for the collaborative and intellectually stimulating environment fostered by all math staff and all my fellow graduate students in the Mathematics Department. I would like to acknowledge Alvarado Taylor, Aldo Garcia Guinto, Bhusal Gokul, Ekblad Owen, Krupansky Nick, Kimble Jamie, Mandela Quashie, Stephen White, Whiting David, Yu Shen, Jie Yang, Peikai Qi, Shitan Xu, Boahen Edem, and many other remarkable peers, whose camaraderie and support have profoundly enriched my Ph.D journey.

I would also like to express my heartfelt gratitude to my mentors at Los Alamos National Laboratory, Dr. Charles Abolt and Dr. Adam Atchley, for their warm invitation, invaluable guidance during my summer internship, and their proactive efforts to extend the opportunity for me to continue contributing part-time afterward. Their support and encouragement have been instrumental in my professional growth.

Lastly but not the least, I would like to express my deepest gratitude to my family—my parents, Gang Su and Fang Wang, and my wife, Xuandi—whose unwavering support and encouragement have served as the bedrock of my journey. Your companionship, and unshakable faith in my potential have been a true pillar of support, I owe a special debt of gratitude. Your understanding and encouragement have inspired me to push through obstacles, remain focused, and strive for excellence. Your collective love and sacrifices that have not only made this journey possible but also deeply meaningful, motivating me to turn my aspirations into reality.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

CHAPTER 1

INTRODUCTION

In an era of unprecedented data generation, extracting meaningful insights from complex, high-dimensional, and often incomplete datasets has become a cornerstone of data science [24, 37]. Matrix and tensor analysis, as foundational tools, provide versatile frameworks to address these challenges. Their applications span a diverse range of fields, including image and video processing [102, 73, 66], recommendation systems [74, 98], and scientific simulations [33, 53, 51**?** , 52]. However, despite their versatility, existing methods often struggle in real-world scenarios characterized by noisy, sparsely observed, or intricately structured data. This thesis focuses on developing robust and flexible methodologies for matrix and tensor analysis, aiming to enhance their robustness to noise and outliers, improve their adaptability to diverse applications, and expand their theoretical underpinnings.

The primary focus of this thesis is on Guaranteed Sampling Flexibility for Low-Tubal-Rank Tensor Completion [107], addressing the limitations of conventional sampling strategies, such as Bernoulli [72, 121, 63] and t-CUR sampling [108, 100], which lack adaptability for diverse real-world applications. To overcome these challenges, this project introduces Tensor Cross-Concentrated Sampling (t-CCS), a generalization of the CCS model to higher-order tensors. Complementing this framework is the development of a novel non-convex algorithm, Iterative Tensor CUR Completion (ITCURC), specifically designed for t-CCS-based tensor completion. The project provides rigorous theoretical foundations, including sufficient conditions for low-tubal-rank tensor recovery and a detailed sampling complexity analysis. Extensive evaluations on synthetic and real-world datasets validate the superior performance of t-CCS and ITCURC in terms of accuracy, flexibility, and computational efficiency. This work advances tensor analysis by addressing the challenges of high-dimensional, incomplete, and sparsely observed data.

The second project explores the Robustness of Cross-Concentrated Sampling (CCS) for Matrix Completion [18], a recent method that leverages cross-sectional dependencies to recover missing data. While CCS has shown promise in capturing essential patterns in data matrices, its vulnerability to sparse outliers—a common challenge in real-world datasets—remains an open question. This project introduces the Robust CCS Completion (RCURC) framework, extending CCS to handle

noisy and incomplete data with resilience to outlier corruption. A non-convex iterative algorithm is developed to solve the RCURC problem, and experimental results on synthetic and real-world datasets demonstrate the algorithm's robustness, efficiency, and scalability.

Together, these projects address critical gaps in matrix and tensor analysis, focusing on robustness, flexibility, and efficiency. The methodologies presented in this thesis not only tackle specific challenges but also provide a foundation for addressing a broader class of problems in data science, where noise, sparsity, and high dimensionality are pervasive. By proposing novel frameworks, designing practical algorithms, and establishing comprehensive theoretical insights, this work advances the state of the art in matrix and tensor analysis, paving the way for their application to increasingly complex and diverse data environments.

This thesis is structured as follows: Chapter 2 explores Tensor Cross-Concentrated Sampling and the Iterative Tensor CUR Completion algorithm. Chapter 3 introduces the Robust CCS Completion framework, detailing its methodology and application to matrix completion problems.

CHAPTER 2

GUARANTEED SAMPLING FLEXIBILITY FOR LOW-TUBAL-RANK TENSOR COMPLETION

ABSTRACT

While Bernoulli sampling is extensively studied in the field of tensor completion, and t-CUR sampling provides a way to approximate low-tubal-rank tensors via lateral and horizontal subtensors, both methods lack sufficient flexibility for diverse practical applications. To address this, we introduces Tensor Cross-Concentrated Sampling (t-CCS), an innovative and straightforward sampling model that advances the matrix cross-concentrated sampling concept within a tensor framework. t-CCS effectively bridges the gap between Bernoulli and t-CUR sampling, offering additional flexibility that can lead to computational savings in various contexts. A key aspect of our work is the comprehensive theoretical analysis provided. We establish a sufficient condition for the successful recovery of a low-rank tensor from its t-CCS samples. In support of this, we also develop a theoretical framework validating the feasibility of t-CUR via uniform random sampling and conduct a detailed theoretical sampling complexity analysis for tensor completion problems utilizing the general Bernoulli sampling model. Moreover, we introduce an efficient non-convex algorithm, the Iterative Tensor CUR Completion (ITCURC) algorithm, specifically designed to tackle the unique challenges of t-CCS-based tensor completion. We have intensively tested and validated the effectiveness of the t-CCS model and the ITCURC algorithm across both synthetic and real-world datasets.

2.1 Introduction

A tensor, as a multidimensional generalization of matrix, provides an intuitive representation for handling multi-relational or multi-modal data such as hyperspectral data [16, 131, 142], videos [80, 103], seismic data [42, 95], DNA microarrays [90]. However, in real-world scenarios, it is common to encounter situations where only partial observations of the tensor data are available due to unavoidable or unforeseen circumstances. These limitations can stem from factors such as data collection issues or errors made during data entry by researchers. The problem of recovering the missing data by effectively leveraging the available observations is commonly referred to as the Tensor Completion (TC) problem.

TC is inherently complex and often ill-posed [49, 141], necessitating the exploration of vari-

ous sampling models and completion techniques. A common and crucial assumption for resolving TC is the low-rank structure of the tensor, which has been extensively utilized to enhance TC approaches [5, 80, 138]. However, the concept of tensor rank is not unique and comes with its own limitations. For example, the CANDECOMP / PARAFAC (CP) rank represents the minimum number of rank-one tensors required to achieve the CP decomposition, involving summations of these tensors [60]. Computing the CP rank–an NP-hard problem–presents difficulties in the recovery of tensors with a low CP rank [68]. Thus, finding the optimal low-CP-rank approximation of the target tensor is still an open problem [141]. Other tensor ranks, such as Tucker [117], Tensor Train [91], tubal [69] and Hierarchical-Tucker [48, 54], to name a few, also play prominent roles in the field, each with its distinct computation and application implications.

In this study, we focus on the low-tubal-rank model for tensor completion. The tubal-rank is defined based on the tensor decomposition known as tensor Singular Value Decomposition (t-SVD), which employs the tensor-tensor product (t-product) [71]. In t-SVD, a tensor is decomposed into the t-product of two orthogonal tensors and a f-diagonal tensor. The tubal-rank is then determined by the number of non-zero singular tubes present in the f-diagonal tensor. Previous research has shown that tubal-rank-based tensor models exhibit better modeling capabilities compared to other rank-based models, particularly for tensors with fixed orientation or specific spatial-shifting characteristics [96, 133]. In low-tubal-rank TC model, we consider $\mathcal{T} \in \mathbb{K}^{n_1 \times n_2 \times n_3}$ with tubal rank r and the observations are located in the set Ω. TC aims to recover the original tensor $\mathcal{T}$ from the observations on Ω. Mathematically, we aim to solve the following optimization problem:

$$\min_{\widetilde{\mathcal{T}}} \langle \mathcal{P}_\Omega(\mathcal{T} - \widetilde{\mathcal{T}}), \mathcal{T} - \widetilde{\mathcal{T}} \rangle, \quad \text{subject to tubal-rank}(\widetilde{\mathcal{T}}) = r, \tag{2.1}$$

where $\langle \cdot, \cdot \rangle$ denotes the Frobenius inner product and $\mathcal{P}_\Omega$ is the sampling operator defined by

$$\mathcal{P}_\Omega(\mathcal{T}) = \sum_{(i,j,k)\in\Omega} [\mathcal{T}]_{i,j,k} \mathcal{E}_{i,j,k} \tag{2.2}$$

where $\mathcal{E}_{i,j,k} \in \{0,1\}^{n_1 \times n_2 \times n_3}$ is a tensor with all elements being zero except for the element at the position indexed by (i, j, k).

For successful recovery, the general setting of an efficient solver for (2.1) requires the observation set Ω to be sampled entry-wise, fiber-wise, or slab-wise through a certain unbiased stochastic process, including the Bernoulli sampling process as referenced in [101, 104, 113, 120] and the uniform sampling process as referenced in [62, 105, 138]. Although extensive theoretical and empirical studies have been conducted on these sampling settings, their practical applicability is sometimes limited in certain contexts. For instance, in collaborative filtering applications, each dimension of the three order tensor data typically represents users, rated items (such as movies or products), and time respectively. The unbiased sampling models implicitly assume that all users are equally likely to rate all items over time, a premise that is often unrealistic in real-world scenarios. Let's consider the application of Magnetic Resonance Imaging (MRI) as another example. MRI scans face limitations with certain metal implants and can cause discomfort in prolonged sessions [1]. To address these issues, we propose a generalization of the cross-concentrated sampling model to the tensor completion setting based on the cross-concentrated sampling model for matrix completion [19], termed tensor cross-concentrated sampling (t-CCS). t-CCS enables partial observations on selected horizontal and lateral subtensors, making it more practical in many applications.

2.1.1 Basic Definitions and Terminology

We use $\mathbb{K}$ to denote an algebraically closed field, either $\mathbb{R}$ or $\mathbb{C}$. We represent a matrix as a capital italic letter (e.g., A) and a tensor by a cursive italic letter (e.g., $\mathcal{T}$). The notation $[n]$ denotes the set of the first n positive integers, i.e., $\{1, \cdots, n\}$, for any $n \in \mathbb{Z}^+$. Submatrices and subtensors are denoted as $[A]_{I,J}$ and $[\mathcal{T}]_{I,J,K}$, respectively, with I, J, K as subsets of appropriate index sets. In particular, if I is the full index set, we denote $[\mathcal{T}]_{:,J,K}$ as $[\mathcal{T}]_{I,J,K}$, and similar rules apply to J and K. Additionally, $|S|$ denotes the cardinality of the set S. If I is a subset of the set $[n]$, then $I^{\complement}$ denotes the set of elements in $[n]$ that are not in I. For a given matrix A, we use $A^\dagger$ to denote its Moore-Penrose inverse and $A^\top$ for its conjugate transpose. The spectral norm of A, represented by $\|A\|$, is its largest singular value. Additionally, the Frobenius norm of A is denoted by $\|A\|_{\mathrm{F}}$, where $\|A\|_{\mathrm{F}} = \sqrt{\sum_{i,j} |A_{i,j}|^2}$, and its nuclear norm, represented by $\|A\|_*$, is the sum of all its singular values.

The Kronecker product is denoted by $\otimes$. The column vector $\mathbf{e}_i$ has a 1 in the i-th position, with other elements as 0, and its dimension is specified when used. For a tensor $\mathcal{T} \in \mathbb{K}^{n_1 \times n_2 \times n_3}$, $\widehat{\mathcal{T}}$ represents the tensor after applying a discrete Fourier transform along its third dimension. Given a tensor $\mathcal{T} \in \mathbb{K}^{n_1 \times n_2 \times n_3}$, we call $[\mathcal{T}]_{i,:,:}$, $[\mathcal{T}]_{:,j,:}$, $[\mathcal{T}]_{:,:,k}$ horizontal, lateral, and frontal slice of $\mathcal{T}$ for any $i \in [n_1]$, $j \in [n_2]$, and $k \in [n_3]$. Figure 2.1 gives an example of the horizontal, lateral, frontal slice of a tensor $\mathcal{T} \in \mathbb{K}^{n_1 \times n_2 \times n_3}$.

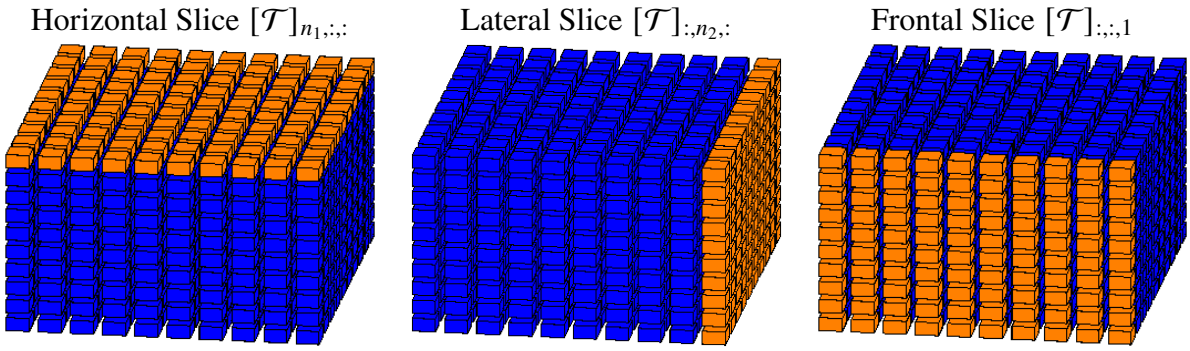


Figure 2.1 Visualization of a horizontal, a lateral, and a frontal slice of a tensor $\mathcal{T} \in \mathbb{K}^{n_1 \times n_2 \times n_3}$. The orange region from the leftmost subfigure, middle subfigure, and rightmost subfigure are a horizontal slice, a lateral slice, and a frontal slice of $\mathcal{T}$ respectively.

Given $\mathcal{T} \in \mathbb{K}^{n_1 \times n_2 \times n_3}$, one can define the associated block circulant matrix obtained from the mode-3 slabs of $\mathcal{T}$ i.e.,

$$\text{bcirc}(\mathcal{T}) := \begin{bmatrix} \mathcal{T}_1 & \mathcal{T}_{n_3} & \cdots & \mathcal{T}_2 \\ \mathcal{T}_2 & \mathcal{T}_1 & \cdots & \mathcal{T}_3 \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{T}_{n_3} & \mathcal{T}_{n_3-1} & \cdots & \mathcal{T}_1 \end{bmatrix} \in \mathbb{K}^{n_1 n_3 \times n_2 n_3},$$

where $\mathcal{T}_i := [\mathcal{T}]_{:,:,i}$. For the purpose of this section, we will utilize a slight modification of the unfolding of a matrix along its second mode and we define

$$\text{unfold}(\mathcal{T}) := \begin{bmatrix} \mathcal{T}_1^\top & \cdots & \mathcal{T}_{n_3}^\top \end{bmatrix}^\top \in \mathbb{K}^{n_1 n_3 \times n_2} \text{ and } \text{fold}(\text{unfold}(\mathcal{T})) = \mathcal{T}.$$

The t-product of tensors $\mathcal{T} \in \mathbb{K}^{n_1 \times n_2 \times n_3}$ and $\mathcal{S} \in \mathbb{K}^{n_2 \times n_4 \times n_3}$ is denoted by $\mathcal{T} * \mathcal{S}$ which is a tensor of dimension $n_1 \times n_4 \times n_3$ obtained via circular convolution. Specifically,

$$\mathcal{T} * \mathcal{S} = \text{fold}(\text{bcirc}(\mathcal{T}) \cdot \text{unfold}(\mathcal{S})). \tag{2.3}$$

The computational cost of the t-product based on Equation (2.3) is $O(n_1 n_2 n_3^2 n_4)$, since $\text{bcirc}(\mathcal{T}) \cdot \text{unfold}(\mathcal{S})$ is the multiplication of a $n_1 n_3 \times n_2 n_3$ matrix with a $n_2 n_3 \times n_4$ matrix.

Define $\overline{\mathcal{T}}$ as

$$\overline{\mathcal{T}} = \left(F_{n_3} \otimes I_{n_1}\right) \cdot \text{bcirc}(\mathcal{T}) \cdot \left(F_{n_3}^{-1} \otimes I_{n_2}\right),$$

where F_n to represents the $n \times n$ Discrete Fourier Transform matrix and F_n^{-1} is its matrix inverse. By the property that a circulant matrix can be block-diagonalized by DFT, we can see that $\overline{\mathcal{T}}$ is a block-diagonal matrix. Notice that

$$\begin{aligned}
\text{unfold}(\mathcal{S}) &= \begin{bmatrix} \mathcal{S}_1 \\ \mathcal{S}_2 \\ \vdots \\ \mathcal{S}_{n_3} \end{bmatrix} \text{(where } \mathcal{S}_i = [\mathcal{S}]_{:,:,i}\text{)} \\
&= \begin{bmatrix} \mathcal{S}_1 & \mathcal{S}_{n_3} & \cdots & \mathcal{S}_2 \\ \mathcal{S}_2 & \mathcal{S}_1 & \cdots & \mathcal{S}_3 \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{S}_{n_3} & \mathcal{S}_{n_3-1} & \cdots & \mathcal{S}_1 \end{bmatrix} \cdot \begin{bmatrix} I_{n_4} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \text{(where } I_{n_4} \text{ is the } n_4 \times n_4 \text{ identity matrix)}
\end{aligned}$$

For simplicity, denote E_1 as $\begin{bmatrix} I_{n_4} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$. Hence, $\mathcal{T} * \mathcal{S}$ can also be expressed as

$$\begin{aligned}
\mathcal{T} * \mathcal{S} &= \text{fold}(\text{bcirc}(\mathcal{T}) \cdot \text{unfold}(\mathcal{S})) \\
&= \text{fold}\left(\text{bcirc}(\mathcal{T}) \cdot \text{bcirc}(\mathcal{S}) \cdot E_1\right) \\
&= \text{fold}\left(\left(F_{n_3}^{-1} \otimes I_{n_1}\right) \cdot \left(F_{n_3} \otimes I_{n_1}\right) \cdot \text{bcirc}(\mathcal{T}) \cdot \left(F_{n_3}^{-1} \otimes I_{n_2}\right) \cdot \left(F_{n_3} \otimes I_{n_2}\right) \cdot \text{bcirc}(\mathcal{S}) \cdot E_1\right) \\
&= \text{fold}\left(\left(F_{n_3}^{-1} \otimes I_{n_1}\right) \cdot \overline{\mathcal{T}} \cdot \left(F_{n_3} \otimes I_{n_2}\right) \cdot \text{bcirc}(\mathcal{S}) \cdot \left(F_{n_3}^{-1} \otimes I_{n_1}\right) \cdot \left(F_{n_3} \otimes I_{n_1}\right) \cdot E_1\right) \\
&= \text{fold}\left(\left(F_{n_3}^{-1} \otimes I_{n_1}\right) \cdot \overline{\mathcal{T}} \cdot \overline{\mathcal{S}} \cdot \left(F_{n_3} \otimes I_{n_1}\right) \cdot E_1\right).
\end{aligned}$$

Numerically, we implement the t-product of two tensors based on Algorithm 2.1.

Algorithm 2.1 t-Product based on Fast Fourier Transform (FFT)

1: Input: $\mathcal{T} \in \mathbb{K}^{n_1 \times n_2 \times n_3}, \mathcal{S} \in \mathbb{K}^{n_2 \times n_4 \times n_3}$.
2: $\mathcal{T} \to \widehat{\mathcal{T}} := \text{fft}(\mathcal{T}, [], 3); \mathcal{S} \to \widehat{\mathcal{S}} := \text{fft}(\mathcal{S}, [], 3)$.
3: **for** each $i \in \{1, 2, \ldots, n_3\}$ **do**
4: $\quad [\widehat{C}]_{:,:,i} = [\widehat{\mathcal{T}}]_{:,:,i} \cdot [\widehat{\mathcal{S}}]_{:,:,i}$.
5: **end for**
6: Output: $\mathcal{Z} \to \text{ifft}(\widehat{\mathcal{Z}}, [], 3)$.

Note that the computational costs of $\text{fft}(\mathcal{T}, [], 3)$, $\text{fft}(\mathcal{S}, [], 3)$ and $\text{ifft}(C, [], 3)$ are $n_1 n_2 n_3 \log(n_3)$, $n_2 n_4 n_3 \log(n_3)$ and $n_1 n_4 n_3 \log(n_3)$ respectively. Thus, t-product based on FFT takes $O(n_1 n_2 n_3 \log(n_3) + n_2 n_4 n_3 \log(n_3) + n_1 n_4 n_3 \log(n_3) + n_1 n_2 n_4 n_3) = O(n_1 n_2 n_4 n_3)$, which is more computational efficient.

Definition 1 (Tensor Frobenius norm)**.** *The tensor Frobenius norm* $\|\mathcal{T}\|_{\text{F}}$ *of a third-order tensor* $\mathcal{T} \in \mathbb{K}^{n_1 \times n_2 \times n_3}$ *is defined as ,*

$$\|\mathcal{T}\|_{\text{F}} := \sqrt{\sum_{i,j,k} |\mathcal{T}_{i,j,k}|^2} = \frac{1}{\sqrt{n_3}} \|\text{bcirc}(\mathcal{T})\|_{\text{F}}.$$

Before we introduce the tensor spectral norm, let's discuss mathematical insight to define such a mathematical object. Suppose that $(V, \|\cdot\|_V)$ and $(W, \|\cdot\|_W)$ are two finite-dimensional linear normed space, where $\|\cdot\|_V$ and $\|\cdot\|_W$ are two norms defined on V and W respectively. Suppose $L : V \to W$ be a continuous linear operator. The operator norm of L can be defined as

$$\|L\| = \sup_{\|v\|_V \leq 1} \|L(v)\|_W.$$

Let $V = \mathbb{K}^{n_1}$ and $W = \mathbb{K}^{n_2}$. Given a matrix $A \in \mathbb{K}^{n_1 \times n_2}$, it is easy to see that operator L defined as

$$\begin{aligned} L : (V, \|\cdot\|_V) &\longrightarrow (W, \|\cdot\|_W) \\ v &\longmapsto L(v) = A \cdot v \end{aligned} \tag{2.4}$$

is a continuous linear operator. Different choices of $\|\cdot\|_V$ and $\|\cdot\|_W$ will lead to different matrix norms induced by the operator norm. For example, if $\|\cdot\|_V$ and $\|\cdot\|_W$ are both Frobenius norm, then the operator norm defined in Equation (2.5) is the same as the matrix spectral norm. Now let's suppose $V = \mathbb{K}^{n_2 \times 1 \times n_3}$ and $W = \mathbb{K}^{n_1 \times 1 \times n_3}$. Given a tensor $\mathcal{A} \in \mathbb{K}^{n_1 \times n_2 \times n_3}$, define operator $\mathcal{L}$ as

$$\begin{aligned} \mathcal{L} : (V, \|\cdot\|_{\text{F}}) &\longrightarrow (W, \|\cdot\|_{\text{F}}) \\ \mathcal{V} &\longmapsto L(\mathcal{V}) = \mathcal{A} * \mathcal{V}. \end{aligned} \tag{2.5}$$

It is easy to check the operator $\mathcal{L}$ is bounded and linear. The operator norm of $\mathcal{L}$ can be computed as follows.

$$\begin{aligned}\|\mathcal{L}\| &:= \sup_{\|\mathcal{V}\|\le 1} \|\mathcal{A} * \mathcal{V}\|_{\mathrm{F}} \\ &= \sup_{\|\mathcal{V}\|_{\mathrm{F}}\le 1} \|\mathrm{bcirc}(\mathcal{A}) \cdot \mathrm{unfold}(\mathcal{V})\|_{\mathrm{F}} \\ &= \|\,\mathrm{bcirc}(\mathcal{A})\|\end{aligned}$$

Remember that $\overline{\mathcal{A}} = \left(F_{n_3} \otimes I_{n_1}\right) \cdot \mathrm{bcirc}(\mathcal{A}) \cdot \left(F_{n_3}^{-1} \otimes I_{n_2}\right)$. Notice that $\frac{1}{\sqrt{n_3}}(F_{n_3} \otimes I_{n_1})^\top \cdot \frac{1}{\sqrt{n_3}}(F_{n_3} \otimes I_{n_1}) = \mathbb{I}_{n_1 n_3}$ and $\sqrt{n_3}(F_{n_3}^{-1} \otimes I_{n_2})^\top \cdot \sqrt{n_3}(F_{n_3}^{-1} \otimes I_{n_2}) = \mathbb{I}_{n_2 n_3}$. Thus, $\frac{1}{\sqrt{n_3}}(F_{n_3} \otimes I_{n_1})$ and $\sqrt{n_3}(F_{n_3}^{-1} \otimes I_{n_2})$ are two unitary orthogonal matrices. Hence, we have

$$\begin{aligned}\|\overline{\mathcal{A}}\| &= \|\left(F_{n_3} \otimes I_{n_1}\right) \cdot \mathrm{bcirc}(\mathcal{A}) \cdot \left(F_{n_3}^{-1} \otimes I_{n_2}\right)\| \\ &= \|\frac{1}{\sqrt{n_3}}(F_{n_3} \otimes I_{n_1})^\top \cdot \left(F_{n_3} \otimes I_{n_1}\right) \cdot \mathrm{bcirc}(\mathcal{A}) \cdot \left(F_{n_3}^{-1} \otimes I_{n_2}\right) \cdot \sqrt{n_3}(F_{n_3}^{-1} \otimes I_{n_2})^\top\| \\ &= \|(F_{n_3} \otimes I_{n_1})^\top \cdot \left(F_{n_3} \otimes I_{n_1}\right) \cdot \mathrm{bcirc}(\mathcal{A}) \cdot \left(F_{n_3}^{-1} \otimes I_{n_2}\right) \cdot (F_{n_3}^{-1} \otimes I_{n_2})^\top\| \\ &= \|(F_{n_3}^\top \otimes I_{n_1}) \cdot \left(F_{n_3} \otimes I_{n_1}\right) \cdot \mathrm{bcirc}(\mathcal{A}) \cdot \left(F_{n_3}^{-1} \otimes I_{n_2}\right) \cdot ((F_{n_3}^{-1})^\top \otimes I_{n_2})\| \\ &= \|n_3(F_{n_3}^{-1} \otimes I_{n_1}) \cdot \left(F_{n_3} \otimes I_{n_1}\right) \cdot \mathrm{bcirc}(\mathcal{A}) \cdot \left(F_{n_3}^{-1} \otimes I_{n_2}\right) \cdot \frac{1}{n_3}(F_{n_3} \otimes I_{n_2})\| \\ &= \|\,\mathrm{bcirc}(\mathcal{A})\|.\end{aligned}$$

Definition 2 (f-diagonal tensor)**.** *A tensor is called f-diagonal if each of its frontal slices is a diagonal matrix.*

Definition 3 (Tensor conjugate transpose)**.** *The conjugate transpose of a tensor $\mathcal{T} \in \mathbb{K}^{n_1 \times n_2 \times n_3}$ is the $n_2 \times n_1 \times n_3$ tensor $\mathcal{T}^\top$ obtained by conjugate transposing each of the frontal slice and then reversing the order of the second to last frontal slices.*

Definition 4 (Identity tensor)**.** *The identity tensor $\mathcal{I} \in \mathbb{K}^{n \times n \times n_3}$ is the tensor with the only first frontal slices $[\mathcal{T}]_{:,:,1}$ being the $n \times n$ identity matrix and with other frontal slices $[\mathcal{T}]_{:,:,i}$ are all zeros for $i = 2, \cdots, n_3$.*

Definition 5 (Orthogonal tensor)**.** *If a tensor of size $n \times n \times n_3$ is orthogonal if $\mathcal{T}^\top * \mathcal{T} = \mathcal{I} = \mathcal{T} * \mathcal{T}^\top = \mathcal{I} \in \mathbb{K}^{n \times n \times n_3}$*

Definition 6 (Partially orthogonal tensor)**.** *If a tensor of size $n_1 \times n_2 \times n_3$ is partially orthogonal if $\mathcal{T}^\top * \mathcal{T} = \mathcal{I} \in \mathbb{K}^{n_2 \times n_2 \times n_3}$ or $\mathcal{T} * \mathcal{T}^\top = \mathcal{I} \in \mathbb{K}^{n_1 \times n_1 \times n_3}$*

Definition 7 (Moore-Penrose inverse [71])**.** *$\mathcal{T}^\dagger \in \mathbb{K}^{n_2 \times n_1 \times n_3}$ is said to be the Moore-Penrose inverse of $\mathcal{T} \in \mathbb{K}^{n_1 \times n_2 \times n_3}$, if $\mathcal{T}^\dagger$ satisfies the following four equations,*

$$\mathcal{T} * \mathcal{T}^\dagger * \mathcal{T} = \mathcal{T}, \ \mathcal{T}^\dagger * \mathcal{T} * \mathcal{T}^\dagger = \mathcal{T}^\dagger,$$
$$\left(\mathcal{T} * \mathcal{T}^\dagger\right)^\top = \mathcal{T} * \mathcal{T}^\dagger, \ \left(\mathcal{T}^\dagger * \mathcal{T}\right)^\top = \mathcal{T}^\dagger * \mathcal{T}.$$

Algorithm 2.2 Moore-Penrose inverse

1: Input: $\mathcal{Z} \in \mathbb{K}^{n_1 \times n_2 \times n_3}$.
2: $\mathcal{Z} \to \widehat{\mathcal{Z}} = \text{fft}(\mathcal{Z}, [], 3)$.
3: **for** each $i \in n_3$ **do**
4: $\quad [\widehat{\mathcal{Z}}]^\dagger_{:,:,i} = \text{Moore-Penrose-inverse}([\widehat{\mathcal{Z}}]_{:,:,i})$
5: **end for**
6: Output: $\mathcal{Z}^\dagger = \text{ifft}(\widehat{\mathcal{Z}}^\dagger, [], 3)$

Definition 8 (Tensor spectral norm and condition number)**.** *The tensor spectral norm $\|\mathcal{T}\|_2$ of a third-order tensor $\mathcal{T}$ is defined as $\|\mathcal{T}\|_2 = \|\text{bcirc}(\mathcal{T})\|_2$. The condition number of $\mathcal{T}$ is defined as: $\kappa(\mathcal{T}) = \|\mathcal{T}^\dagger\|_2 \cdot \|\mathcal{T}\|_2$.*

Definition 9 (Standard tensor lateral basis)**.** *The lateral basis $\mathring{\mathbf{e}}_i$, is of size $n_1 \times 1 \times n_3$ with only $[\mathring{\mathbf{e}}_i]_{i,1,1}$ equal to 1 and the remaining equal to zero.*

Definition 10 (Standard tensor tubal basis [109, 138])**.** *A standard tubal basis $\dot{\mathbf{e}}_k$, is a $1 \times 1 \times n_3$ third mode tensor where all elements are zero except for a single nonzero element with a value of 1 at the $(1, 1, k)$ entry.*

Definition 11 (Identity tensor)**.** *The identity tensor $\mathcal{I} \in \mathbb{K}^{n \times n \times n_3}$ is the tensor whose first frontal slice is the $n \times n$ identity matrix and other frontal slices are all zeros.*

Our research will focus on the subtensors of an underlying tensor with low tubal-rank. To ensure that this work is self-contained, we will begin by introducing the concept of the sampling tensor as follows.

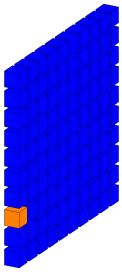


Figure 2.2 A Standard Tensor Lateral Basis.

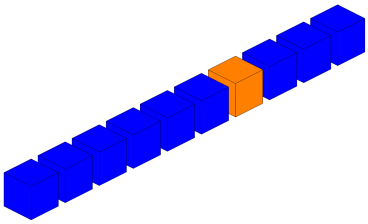


Figure 2.3 A Standard Tubal Basis.

Definition 12 (Sampling tensor)**.** *Given a tensor* $\mathcal{T} \in \mathbb{K}^{n_1 \times n_2 \times n_3}$ *and* $I \subseteq [n_1]$*, the horizontal subtensor* $\mathcal{R}$ *of* $\mathcal{T}$ *with indices* I *can be obtained via*

$$\mathcal{R} := [\mathcal{T}]_{I,:,:} = [\mathcal{I}]_{I,:,:} * \mathcal{T},$$

where $\mathcal{I}$ *is defined in Definition 11.*

For convenience, $[\mathcal{I}]_{I,:,:}$ *will be denoted by* $\mathcal{S}_I$ *with the given index set* I*. Similarly, the lateral sub-tensor* C *with indices* $J \subseteq [n_2]$ *can be obtained as* $C := [\mathcal{T}]_{:,J,:} = \mathcal{T} * [\mathcal{I}]_{:,J,:}$*. The subtensor* $\mathcal{W}$ *of* $\mathcal{T}$ *with horizontal indices* I *and lateral indices* J *can be represented as* $\mathcal{U} := [\mathcal{T}]_{I,J,:} = \mathcal{S}_I * \mathcal{T} * \mathcal{S}_J$*.*

2.1.2 Tensor decomposition

Tensor decompositions provide a concise representation of the underlying structure of data, revealing the low-dimensional subspace within which the data resides.

Theorem 2.1 (t-SVD)**.** *Let* $\mathcal{T} \in \mathbb{K}^{n_1 \times n_2 \times n_3}$*. Then, it can be factored as*

$$\mathcal{T} = \mathcal{W} * \Sigma * \mathcal{V}^\top,$$

where $\mathcal{W} \in \mathbb{K}^{n_1 \times n_1 \times n_3}, \mathcal{V} \in \mathbb{K}^{n_2 \times n_2 \times n_3}$ *are orthogonal and* $\Sigma \in \mathbb{K}^{n_1 \times n_2 \times n_3}$ *is a f-diagonal tensor.*

Numerically, we implement t-SVD based on Algorithm 2.3.

Algorithm 2.3 t-SVD

1: Input: $\mathcal{T} \in \mathbb{K}^{n_1 \times n_2 \times n_3}$.
2: $\mathcal{Z} \to \widehat{\mathcal{Z}} := \text{fft}(\mathcal{Z}, [], 3)$.
3: **for** each $i \in n_3$ **do**
4: $\quad [[\widehat{\mathcal{U}}]_{:,:,i}, [\widehat{\mathcal{S}}]_{:,:,i}, [\widehat{\mathcal{V}}]_{:,:,i}] = \text{SVD}([\widehat{\mathcal{Z}}]_{:,:,i})$
5: **end for**
6: Output: $\text{ifft}(\widehat{\mathcal{U}}, [], 3); \text{ifft}(\widehat{\mathcal{S}}, [], 3); \text{ifft}(\widehat{\mathcal{V}}, [], 3)$

From Algorithm 2.3, we can see that t-SVD is implemented by performing matrix SVD iteratively for a loop of n_3 times. Thus, the computational complexity of t-SVD of a $n_1 \times n_2 \times n_3$ is $O(\min\{n_1^2 n_2 n_3, n_1 n_2^2 n_3\})$

Definition 13 (Tubal-rank and multi-rank)**.** *Suppose the tensor* $\mathcal{T} \in \mathbb{K}^{n_1 \times n_2 \times n_3}$ *satisfies* $\operatorname{rank}\left([\widehat{\mathcal{T}}]_{:,:,k}\right) = r_k$ *for* $k \in [n_3]$*. Then* $\vec{r} = (r_1, r_2, \ldots, r_{n_3})$ *is called the multi-rank of* $\mathcal{T}$*, denoted by* $\operatorname{rank}_m(\mathcal{T})$*. In addition,* $\max\{r_i : i \in [n_3]\}$ *is called the tubal-rank of* $\mathcal{T}$*, denoted by* $\operatorname{rank}(\mathcal{T})$*. We denote tubal-rank as r or* $\|\vec{r}\|_\infty$*, and* $\|\vec{r}\|_1$ *for the sum of the multi-rank.*

Theorem 2.2 (Compact t-SVD)**.** *Let* $\mathcal{T} \in \mathbb{K}^{n_1 \times n_2 \times n_3}$ *with tubal-rank r. Then, it can be factored as*

$$\mathcal{T} = \mathcal{W} * \Sigma * \mathcal{V}^\top,$$

where $\mathcal{W} \in \mathbb{K}^{n_1 \times r \times n_3}, \mathcal{V} \in \mathbb{K}^{n_2 \times r \times n_3}$ *are partially orthogonal and* $\Sigma \in \mathbb{K}^{r \times r \times n_3}$ *is a f-diagonal tensor. Numerically, we implement compact t-SVD based on Algorithm 2.4.*

Algorithm 2.4 Compact t-SVD

Input: $\mathcal{T} \in \mathbb{K}^{n_1 \times n_2 \times n_3}$.
$\mathcal{Z} \to \widehat{\mathcal{Z}} := \text{fft}(\mathcal{Z}, [], 3)$.
Initialize $\widehat{\mathcal{W}} = \text{zeros}(n_1, r, n_3)$, $\widehat{\mathcal{S}} = \text{zeros}(r, r, n_3)$ and $\widehat{\mathcal{V}} = \text{zeros}(n_2, r, n_3)$.
for each $i \in n_3$ **do**
    $[W, S, V] = \text{SVD}([\widehat{\mathcal{Z}}]_{:,:,i})$
    $[\widehat{\mathcal{W}}]_{:,:,i} = [W]_{:,1:r}$;
    $[\widehat{\mathcal{S}}]_{:,:,i} = [S]_{1:r,1:r}$;
    $[\widehat{\mathcal{V}}]_{:,:,i} = [V]_{:,1:r}$
end for
Output: $\mathcal{W} = \text{ifft}(\widehat{\mathcal{W}}, [], 3); \mathcal{S} = \text{ifft}(\widehat{\mathcal{S}}, [], S); \mathcal{V} = \text{ifft}(\widehat{\mathcal{V}}, [], 3)$

Lemma 2.1. *[71, 69] [Best Tubal rank-r approximation]. Let the t-SVD of* $\mathcal{T} \in \mathbb{R}^{m \times n \times k}$ *be* $\mathcal{T} = \mathcal{U} * \mathcal{S} * \mathcal{V}^\dagger$*. For a given positive integer r, define* $\mathcal{T}_r = \sum_{s=1}^{r} \mathcal{U}(:, s, :) * \mathcal{S}(s, s, :) * \mathcal{V}^\dagger(:, s, :)$*. Then* $\mathcal{T}_r = \operatorname{argmin}_{\overline{\mathcal{T}} \in \mathbb{T}} ||\mathcal{T} - \overline{\mathcal{T}}||_F$*, where* $\mathbb{T} = \{\mathcal{X} * \mathcal{Y}^\dagger | \mathcal{X} \in \mathbb{K}^{m \times r \times k}, \mathcal{Y} \in \mathbb{K}^{n \times r \times k}\}$.

Note that $\mathcal{S}$ in t-SVD is organized in a decreasing order, i.e., $||\mathcal{S}(1, 1, :)||_2 \geq ||\mathcal{S}(2, 2, :)||_2 \geq \ldots$, which is implicitly defined in [69]. Therefore, the best rank-r approximation of tensors is similar to PCA (principal component analysis) of matrices. After we introduce the compact t-SVD, we introduce two important definitions based on this type of decomposition.

Definition 14 (Tensor μ_0-incoherence condition)**.** *Given a tubal-rank r tensor $\mathcal{T} \in \mathbb{K}^{n_1 \times n_2 \times n_3}$ with a compact t-SVD $\mathcal{T} = \mathcal{W} * \mathcal{S} * \mathcal{V}^\top$, we say $\mathcal{T}$ satisfy μ_0-incoherence condition if for all $k \in \{1 \cdots, n_3\}$, the following hold:*

$$\max_{i=1,2,.,n_1} \left\| [\widehat{\mathcal{W}}]_{:,:,k}^\top \cdot \mathbf{e}_i \right\|_{\mathrm{F}} \leq \sqrt{\frac{\mu_0 r}{n_1}}, \max_{j=1,2,\ldots,n_2} \left\| [\widehat{\mathcal{V}}]_{:,:,k}^\top \cdot \mathbf{e}_j \right\|_{\mathrm{F}} \leq \sqrt{\frac{\mu_0 r}{n_2}}.$$

In certain instances, to accentuate the incoherence parameter of a specific tensor $\mathcal{T}$, we will represent this parameter as $\mu_{\mathcal{T}}$.

In tensor decomposition, t-CUR decomposition, a self-expressiveness tensor decomposition of a given 3-mode tensor, has received significant attention [4, 28, 56, 109]. Specifically, t-CUR involves representing a tensor $\mathcal{T} \in \mathbb{K}^{n_1 \times n_2 \times n_3}$ as $\mathcal{T} \approx \mathcal{C} * \mathcal{U} * \mathcal{R}$, with $\mathcal{C} = [\mathcal{T}]_{:,J,:}$ and $\mathcal{R} = [\mathcal{T}]_{I,:,:}$ for some $J \subseteq [n_2]$ and $I \subseteq [n_1]$. There exist different versions of $\mathcal{U}$. This work focuses on the t-CUR decomposition of the form $\mathcal{T} \approx \mathcal{C} * \mathcal{U}^\dagger * \mathcal{R}$ with $\mathcal{U} = [\mathcal{T}]_{I,J,:}$. Under certain conditions, this approximation accurately represents $\mathcal{T}$. [28, 56] have detailed the conditions for exact t-CUR decomposition. We begin by defining the tubal-rank of a 3-mode tensor:

For convenience, we present one theoretical result of t-CUR below.

Theorem 2.3 ([28, 56])**.** *Let $\mathcal{T} \in \mathbb{K}^{n_1 \times n_2 \times n_3}$ with multi-rank* $\mathrm{rank}_m(\mathcal{T}) = \vec{r}$. *$I \subset [n_1]$ and $J \subset [n_2]$ are two index sets. Denote $\mathcal{C} = [\mathcal{T}]_{:,J,:}$, $\mathcal{R} = [\mathcal{T}]_{I,:,:}$, and $\mathcal{U} = [\mathcal{T}]_{I,J,:}$. Then $\mathcal{T} = \mathcal{C} * \mathcal{U}^\dagger * \mathcal{R}$ if and only if* $\mathrm{rank}_m(\mathcal{C}) = \mathrm{rank}_m(\mathcal{R}) = \vec{r}$.

Theorem 2.3 can be visualized as follows.

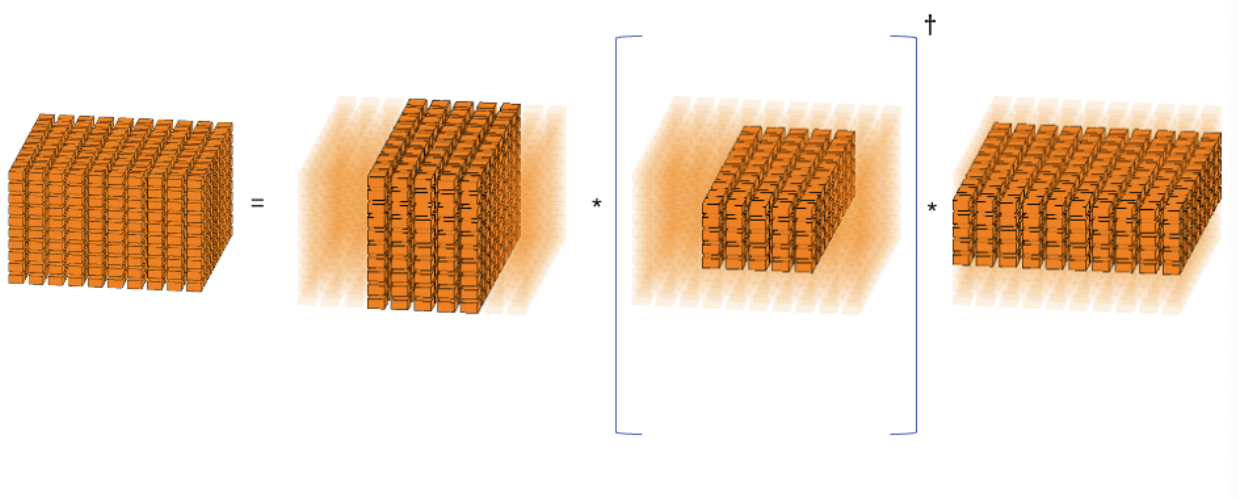


Figure 2.4 t-CUR decomposition.

2.1.3 Related work

Kilmer and Martin [71] introduced novel definitions for tensor multi-rank and tubal-rank characterized by the t-SVD. Researchers commonly employ a convex surrogate to tubal-rank function augmented with regularization of the tensor nuclear norm (TNN), as indicated in [64, 66, 82, 84, 138, 143]. While a pioneering optimization method featuring TNN is initially proposed to tackle the TC problem in [139], this approach necessitates the simultaneous minimization of all singular values across tensor slices, which hinders its ability to accurately approximate the tubal-rank function [61, 130]. To circumvent this challenge, various truncated methods have been introduced as alternatives. Notably, examples include the truncated nuclear norm regularization [61] and the tensor truncated nuclear norm (T-TNN) [130]. Furthermore, Zhang et al. [135] introduced a novel strategy for low-rank regularization, focusing on nonlocal similar patches. However, the aforementioned tensor completion algorithms are designed based on Bernoulli sampling model. Despite its foundational role in probability theory and statistics, Bernoulli sampling frequently encounters practical limitations when applied to real-world data collection scenarios [43, 94]. In the realm of collaborative filtering, where the tensor's horizontal and lateral slices denote users and rated objects (such as movies and merchandise) over a specific time period, the application of the Bernoulli sampling model is impractical. As this model implicitly assumes that every user has an equal prob-

ability of rating any given object, an assumption that is seldom valid in real-world scenarios. The variability in user preferences and interaction patterns makes this equal-probability assumption unrealistic, thereby challenging the efficacy of the Bernoulli sampling approach in such contexts.

2.2 Proposed sampling model

We aim to develop a sampling strategy that is both efficient and effective for a range of real-world scenarios. Inspired by the cross-concentrated sampling model for matrix completion [19] and t-CUR decomposition [28, 56, 109], we introduce a novel sampling model tailored for tensor data, named Tensor Cross-Concentrated Sampling (t-CCS). The t-CCS model extracts samples from both horizontal and lateral subtensors of the original tensor. Formally, let $\mathcal{R} = [\mathcal{T}]_{I,:,:}$ and $C = [\mathcal{T}]_{:,J,:}$ be the selected horizontal and lateral subtensors of $\mathcal{T}$, determined by index sets I and J respectively. Next, we sample entries on $\mathcal{R}$ and C based on the Bernoulli sampling model. The t-CCS procedure is detailed in Procedure 2.5. Notably, t-CCS transitions to t-CUR sampling when the samples are dense enough to fully capture the subtensors and reverts to Bernoulli sampling when all horizontal and lateral slices are selected. The indices of the cross-concentrated samples are denoted as $\Omega_{\mathcal{R}}$ and Ω_C, corresponding to the notation used for the subtensors. Our task is to recover an underlying tensor $\mathcal{T}$ with tubal-rank r from the observations on $\Omega_{\mathcal{R}} \cup \Omega_C$:

$$\min_{\widetilde{\mathcal{T}}} \left\langle \mathcal{P}_{\Omega_{\mathcal{R}} \cup \Omega_C}(\mathcal{T} - \widetilde{\mathcal{T}}), \mathcal{T} - \widetilde{\mathcal{T}} \right\rangle, \quad \text{subject to} \quad \text{tubal-rank}(\widetilde{\mathcal{T}}) = r, \tag{2.6}$$

where $\langle \cdot, \cdot \rangle$ is the Frobenius inner product and $\mathcal{P}_{\Omega_{\mathcal{R}} \cup \Omega_C}$ is defined in (2.2).

Algorithm 2.5 Tensor Cross-Concentrated Sampling (t-CCS)

1: **Input:** $\mathcal{T} \in \mathbb{K}^{n_1 \times n_2 \times n_3}$.
2: Uniformly select the horizontal and lateral indices, denoted as I and J, respectively.
3: Set $\mathcal{R} := [\mathcal{T}]_{I,:,:}$ and $C := [\mathcal{T}]_{:,J,:}$.
4: Sample entries from $\mathcal{R}$ and C based on Bernoulli sampling models. Record the locations of these samples as $\Omega_{\mathcal{R}}$ and Ω_C for $\mathcal{R}$ and C, respectively.
5: **Output:** $[\mathcal{T}]_{\Omega_{\mathcal{R}} \cup \Omega_C}$, $\Omega_{\mathcal{R}}$, Ω_C, I, J.

This chapter aims to provides a theoretical well-posedness of the t-CCS model, our key theoretical contribution, detailed in Theorem 2.6.

2.3 Theoretical Results

This section is dedicated to providing a theoretical analysis of the well-posedness of the t-CCS model, which represents our principal theoretical contribution. This analysis is thoroughly detailed in Theorem 2.6. Before presenting our main theoretical result Theorem 2.6, we firstly introduce two important supporting theorems, where the proof of main theoretical result rely on. The first theorem, Theorem 2.4, establishes the necessary lower bounds for the number of lateral and horizontal slices required when uniformly sampling these slices to ensure an exact t-CUR decomposition. Theorem 2.4 can be seen as an adaptation of [109, Corollary 3.10], featuring a different proof method specifically designed for uniform sampling and exact t-CUR, and provides a more thorough analysis for this specific context.

Before presenting Theorem 2.4, let's briefly review the sampling schemes for matrix CUR decomposition. Various sampling schemes are designed to ensure the chosen rows and columns validate the CUR decomposition. For example, deterministic methods are explored in works such as [6, 8, 79]. Randomized sampling algorithms for CUR decompositions and the column subset selection problem have been extensively studied, as seen in [32, 38, 40, 86, 114, 122]. For a comprehensive overview of both approaches, refer to [57]. Hybrid methods that combine both approaches are discussed in [9, 10, 17].

Particularly, for a rank r matrix in $\mathbb{K}^{n_1 \times n_2}$ with μ-incoherence, sampling $O(\mu r \log(n_1))$ rows and $O(\mu r \log(n_2))$ columns is sufficient to ensure the exact matrix CUR decomposition [12, 32]. In this work, we extend the uniform sampling results from the matrix setting to the tensor setting.

Theorem 2.4. *Let $\mathcal{T} \in \mathbb{K}^{n_1 \times n_2 \times n_3}$ satisfy the tensor μ_0-incoherence condition and have multi-rank $\vec{r}$. The indices I and J are selected uniformly randomly without replacement from $[n_1]$ and $[n_2]$ respectively. Set $\mathcal{C} = [\mathcal{T}]_{:,J,:}$, $\mathcal{R} = [\mathcal{T}]_{I,:,:}$ and $\mathcal{U} = [\mathcal{T}]_{I,J,:}$. Then $\mathcal{T} = \mathcal{C} * \mathcal{U}^{\dagger} * \mathcal{R}$ holds with*

probability at least $1 - \frac{1}{n_1^\beta} - \frac{1}{n_2^\beta}$, *provided that*

$$|I| \geq 2\beta\mu_0\|\vec{r}\|_\infty \log\left(n_1\|\vec{r}\|_1\right)$$
$$|J| \geq 2\beta\mu_0\|\vec{r}\|_\infty \log\left(n_2\|\vec{r}\|_1\right).$$

Another important supporting theorem is Theorem 2.5, which adapts [138, Theorem 3.1] for tensor recovery with tubal-rank r under Bernoulli sampling, essential for Theorem 2.6. Our contribution refines the theorem by explicitly detailing the numerical constants in the original sampling probability. The proof of Theorem 2.5 is under the same framework as in [84, 138].

Theorem 2.5. *Let* $\mathcal{T} \in \mathbb{K}^{n_1\times n_2\times n_3}$ *of tubal-rank* r *satisfy the tensor* μ_0*-incoherence condition. And its compact t-SVD is* $\mathcal{T} = \mathcal{U} * \mathcal{S} * \mathcal{V}^\top$ *where* $\mathcal{U} \in \mathbb{K}^{n_1\times r\times n_3}, \mathcal{S} \in \mathbb{K}^{r\times r\times n_3}$ *and* $\mathcal{V} \in \mathbb{K}^{n_2\times r\times n_3}$. *Suppose the entries in* Ω *are sampled according to the Bernoulli model with probability* p. *If*

$$p \geq \frac{256\beta(n_1+n_2)\mu_0 r\log^2(n_1n_3+n_2n_3)}{n_1n_2} \text{ with } \beta \geq 1, \tag{2.7}$$

then $\mathcal{T}$ *is the unique minimizer to*

$$\min_{\mathcal{T}} \|\mathcal{T}\|_{\text{TNN}}, \text{ subject to } \mathcal{P}_\Omega(\mathcal{T}) = \mathcal{P}_\Omega(\mathcal{T}),$$

with probability at least $1 - \frac{3\log(n_1n_3+n_2n_3)}{(n_1n_3+n_2n_3)^{4\beta-2}}$.

Theorem 2.6. *Let* $\mathcal{T} \in \mathbb{K}^{n_1\times n_2\times n_3}$ *satisfy the tensor* μ_0*-incoherence condition and have multi-rank* $\vec{r}$ *with condition number* κ. *Let* $I \subseteq [n_1], J \subseteq [n_2]$ *be chosen uniformly with replacement to yield* $\mathcal{R} = [\mathcal{T}]_{I,:,:}$ *and* $C = [\mathcal{T}]_{:,J,:}$. *And suppose that* $\Omega_\mathcal{R}$ *and* Ω_C *are generated from* $\mathcal{R}$ *and* C *according to the Bernoulli distributions with probability* $p_\mathcal{R}$ *and* p_C *respectively. If*

$$|I| \geq 3200\beta\mu_0 r\kappa^2\log^2(n_1n_3+n_2n_3),$$
$$|J| \geq 3200\beta\mu_0 r\kappa^2\log^2(n_1n_3+n_2n_3),$$
$$p_\mathcal{R} \geq \frac{1600(|I|+n_2)\mu_0 r\kappa^2\log^2((n_1+n_2)n_3)}{|I|n_2},$$
$$p_C \geq \frac{1600(|J|+n_1)\mu_0 r\kappa^2\log^2((n_1+n_2)n_3)}{|J|n_1}$$

for some absolute constant $\beta > 1$, then $\mathcal{T}$ can be uniquely determined from the entries on $\Omega_{\mathcal{R}} \cup \Omega_C$ with probability at least

$$1 - \frac{1}{(n_1n_3 + n_2n_3)^{800\beta\kappa^2 \log(n_2)}} - \frac{1}{(n_1n_3 + n_2n_3)^{800\beta\kappa^2 \log(n_1)}} - \frac{3\log(n_1n_3 + |J|n_3)}{(n_1n_3 + |J|n_3)^{4\beta-2}} - \frac{3\log(n_2n_3 + |I|n_3)}{(n_2n_3 + |I|n_3)^{4\beta-2}}.$$

Remark 1. *(i) When $n_1 = n_2 = n$, the results in the above theorem can be simplified to that $\mathcal{T}$ can be uniquely determined from the entries on $\Omega_{\mathcal{R}} \cup \Omega_C$ with probability at least $1 - \frac{6\log(2nn_3)}{(nn_3)^{4\beta-2}}$.*

(ii) Supposed $\mathcal{T}$ with multi-rank $\vec{r}$ of low tubal-rank r is the underlying tensor we aim to recover. Notice that such $\mathcal{T}$ is one of feasible solutions to the optimization problem (2.1) since tubal-rank$(\mathcal{T}) = r$. Additionally, it is evident that for any $\widetilde{\mathcal{T}}$ with tubal-rank r,

$$\langle \mathcal{P}_\Omega(\widetilde{\mathcal{T}} - \mathcal{T}), \widetilde{\mathcal{T}} - \mathcal{T})\rangle \geq 0 \ \text{ and } \ \langle \mathcal{P}_\Omega(\mathcal{T} - \mathcal{T}), \mathcal{T} - \mathcal{T}\rangle = 0.$$

Thus, $\mathcal{T}$ is a global minimizer to the optimization problem (2.1). According to Theorem 2.6, $\mathcal{T}$ with low tubal-rank r can be reliably recovered using the t-CCS model with high probability. Consequently, we can obtain a minimizer for the non-convex optimization problem (2.1) through samples that are partially observed from the t-CCS model.

Theorem 2.6 elucidates that a sampling complexity of

$$O(r\kappa^2 \max\{n_1, n_2\} n_3 \log^2(n_1n_3 + n_2n_3))$$

is sufficient for TC on t-CCS model. This complexity is a κ^2 factor worse than that of the benchmark provided by the state-of-the-art Bernoulli-sampling-based TC methods, such as the TNN method detailed by Zhang and Aeron [138], which demands

$$O(r \max\{n_1, n_2\} n_3 \log^2(n_1n_3 + n_2n_3))$$

samples. This observation suggests the potential for identifying a more optimal lower bound, which will leave as a future direction.

2.4 An efficient solver for t-CCS

In this section, we investigate how to effectively and efficiently solve the t-CCS-based TC problem. First, we consider directly applying several existing TC algorithms including BCPF [140], TMac [129], TNN [138], and F-TNN [66] to a t-CCS-based image recovery problem, where BCPF is CP-based algorithm, TMac is Tucker-based algorithm, TNN and F-TNN are two tubal-rank-based algorithms. However, it turns out that these methods are not well-suited for the tensor completion problem based on t-CCS model. As illustrated in Figure 2.5, these approaches fail to yield reliable visualization outcomes. This indicates the necessity to develop new algorithm(s) for the proposed t-CCS model.

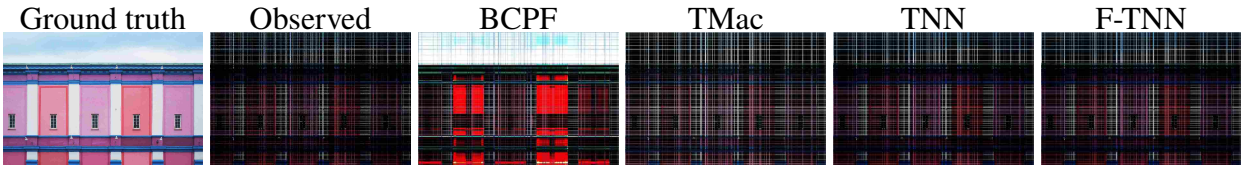


Figure 2.5 Visual results of color image inpainting using t-CCS samples at an overall sampling rate of 20% with BCPF, TMac, TNN, and F-TNN algorithms.

2.4.1 Iterative tensor CUR completion algorithm

To efficiently use the t-CCS structure, we develop the Iterative Tensor CUR Completion (ITCURC), a non-convex algorithm inspired by projected gradient descent. ITCURC updates $\mathcal{R}$, $\mathcal{C}$, and $\mathcal{U}$ at each iteration to preserve the tubal-rank r of $\mathcal{T}$. The update formulas are:

$$[\mathcal{R}_{k+1}]_{:,J^{\complement},:} := [\mathcal{T}_k]_{I,J^{\complement},:} + \eta_R \left[\mathcal{P}_{\Omega_{\mathcal{R}}} (\mathcal{T} - \mathcal{T}_k)\right]_{I,J^{\complement},:}, \tag{2.8}$$

$$[\mathcal{C}_{k+1}]_{I^{\complement},:,:} := [\mathcal{T}_k]_{I^{\complement},J} + \eta_C \left[\mathcal{P}_{\Omega_C} (\mathcal{T} - \mathcal{T}_k)\right]_{I^{\complement},J,:}, \tag{2.9}$$

$$\mathcal{U}_{k+1} := \mathcal{H}_r \left([\mathcal{T}_k]_{I,J,:} + \eta_U \left[\mathcal{P}_{\Omega_R \cup \Omega_C} (\mathcal{T} - \mathcal{T}_k)\right]_{I,J,:} \right), \tag{2.10}$$

where η_R, η_C, η_U are step sizes, and $\mathcal{H}_r$ is the truncated t-SVD operator. $[\mathcal{R}_{k+1}]_{:,J,:}$ and $[\mathcal{C}_{k+1}]_{I,:,:}$ are updated to $\mathcal{U}_{k+1}$. The algorithm, starting from $\mathcal{T}_0 = \mathbf{0}$, iterates until $e_k \leq \varepsilon$, where ε is a preset

tolerance and

$$e_k = \frac{\left\langle \mathcal{P}_{\Omega_\mathcal{R}\cup\Omega_C}\left(\mathcal{T}-\mathcal{T}_k\right), \mathcal{T}-\mathcal{T}_k\right\rangle}{\left\langle \mathcal{P}_{\Omega_\mathcal{R}\cup\Omega_C}(\mathcal{T}), \mathcal{T}\right\rangle}. \tag{2.11}$$

The algorithm is summarized in Algorithm 2.6.

Algorithm 2.6 Iterative CUR tensor completion for t-CCS (ITCURC)

Input: $[\mathcal{T}]_{\Omega_\mathcal{R}\cup\Omega_C}$: observed data; $\Omega_\mathcal{R}$, Ω_C : observed locations; I, J : horizontal and lateral indexes that define $\mathcal{R}$ and C respectively; η_R, η_C, η_U : step sizes; r : target tubal-rank; ε : tolerance level.
Set $\mathcal{T}_0 = 0 \in \mathbb{K}^{n_1\times n_2\times n_3}$.
while $e_k > \varepsilon$ **do** // e_k is defined in (2.11)
  $[\mathcal{R}_{k+1}]_{:,J^\complement,:} = [\mathcal{T}_k]_{I,J^\complement,:} + \eta_R[\mathcal{P}_{\Omega_\mathcal{R}}([\mathcal{T}]_{\Omega_\mathcal{R}\cup\Omega_C} - \mathcal{T}_k)]_{I,J^\complement,:}$
  $[C_{k+1}]_{I^\complement,:,:} = [\mathcal{T}_k]_{I^\complement,J,:} + \eta_C[\mathcal{P}_{\Omega_C}([\mathcal{T}]_{\Omega_\mathcal{R}\cup\Omega_C} - \mathcal{T}_k)]_{I^\complement,J,:}$
  $\mathcal{U}_{k+1} = \mathcal{H}_r([\mathcal{T}_k]_{I,J,:} + \eta_U[\mathcal{P}_{\Omega_\mathcal{R}\cup\Omega_C}([\mathcal{T}]_{\Omega_\mathcal{R}\cup\Omega_C} - \mathcal{T}_k)]_{I,J,:})$
  $[\mathcal{R}_{k+1}]_{:,J,:} = \mathcal{U}_{k+1}$ and $[C_{k+1}]_{I,:,:} = \mathcal{U}_{k+1}$.
  Update $\mathcal{T}_{k+1}$// More details see (2.12), (2.13), (2.14)
  $k = k + 1$
end while
Output: C_{k+1}, $\mathcal{U}_{k+1}$ and $\mathcal{R}_{k+1}$

Now let's outline Algorithm 2.6's implementation and computational costs. Updating $[\mathcal{R}_{k+1}]_{:,J^\complement,:}$ and $[C_{k+1}]_{I^\complement,:,:}$ incurs $O\left(|\Omega_\mathcal{R}| + |\Omega_C| - |\Omega_\mathcal{U}|\right)$ flops, focusing only on observed locations (refer to (2.8) and (2.9)). The update of $\mathcal{U}_{k+1}$, sized $|I| \times |J| \times n_3$, involves (i) computing $\widetilde{\mathcal{U}}_{k+1} := [\mathcal{T}_k]_{I,J,:} + \eta_U[\mathcal{P}_{\Omega_\mathcal{R}\cup\Omega_C}([\mathcal{T}]_{\Omega_\mathcal{R}\cup\Omega_C} - \mathcal{T}_k)]_{I,J,:}$ and (ii) finding its tubal-rank r approximation via t-SVD. The cost for (i) is $O(|\Omega_\mathcal{U}|)$, while (ii) requires $\max\{O(|I||J|rn_3), O(|I||J|n_3\log(n_3))\}$, making the total update cost for $\mathcal{U}_k$ to be $\max\{O(|I||J|rn_3), O(|I||J|n_3\log(n_3))\}$.

Considering the cost of updating $\mathcal{T}_{k+1}$ in Algorithm 2.6, we focus on $[\mathcal{T}_{k+1}]_{I^\complement,J,:}$, $[\mathcal{T}_{k+1}]_{I,J^\complement,:}$, and $[\mathcal{T}_{k+1}]_{I,J,:}$ each iteration. The update for $[\mathcal{T}_{k+1}]_{I^\complement,J,:}$ is:

$$[\mathcal{T}_{k+1}]_{I^\complement,J,:} = [C_k]_{I^\complement,:,:} * \mathcal{U}_k^\dagger * \mathcal{U}_k = [C_k]_{I^\complement,:,:} * [\mathcal{V}_k]_{:,1:r,:} * [\mathcal{V}_k]_{:,1:r,:}^\top. \tag{2.12}$$

where $\mathcal{U}_k = \mathcal{W}_k * \Sigma_k * \mathcal{V}_k^\top$ is $\mathcal{U}_k$'s t-SVD. Given $[\mathcal{V}_k]_{:,1:r,:}$'s size as $|J| \times r \times n_3$, the computational cost is $O(n_1|J|rn_3)$ flops for (2.12), making the total complexity for updating $[\mathcal{T}_{k+1}]_{I^\complement,J,:}$ also $O(n_1|J|rn_3)$ flops. We update $[\mathcal{T}_{k+1}]_{I,J^\complement,:}$ by

$$[\mathcal{T}_{k+1}]_{I,J^\complement,:} := [\mathcal{W}_k]_{:,1:r,:} * [\mathcal{W}_k]_{:,1:r,:}^\top * [\mathcal{R}_k]_{:,J^\complement,:}. \tag{2.13}$$

Similar analysis for updating $[\mathcal{T}_{k+1}]_{I,J^{\complement},:}$, the computational complexity of updating $[\mathcal{T}_{k+1}]_{I,J^{\complement},:}$ is $O(n_2|I|rn_3)$. And we update $[\mathcal{T}_{k+1}]_{I,J,:}$ by setting

$$[\mathcal{T}_{k+1}]_{I,J,:} := \mathcal{U}_k. \tag{2.14}$$

Thus, computational complexity of updating $\mathcal{T}_{k+1}$ is $O(|I|rn_2n_3 + |J|rn_1n_3)$.

Computation of the stopping criterion e_k cost $O\left(|\Omega_R| + |\Omega_C| - |\Omega_U|\right)$ flops as we only make computations on the observed locations.

The computational costs per iteration are summarized in Table 2.1, showing a complexity of $O(r|I|n_2n_3 + r|J|n_1n_3)$ when $|I| \ll n_1$ and $|J| \ll n_2$.

Table 2.1 A Comprehensive Examination of the Per-Iteration Computational Cost for ITCURC.

Step	Computational Complexity
Line 3: Computing the stopping criterion e_k	$O\left(\lvert\Omega_R\rvert + \lvert\Omega_C\rvert - \lvert\Omega_U\rvert\right)$
Line 4: $[\mathcal{R}_{k+1}]_{:,J^{\complement},:} = [\mathcal{T}_k]_{I,J^{\complement},:} + \eta_R[\mathcal{P}_{\Omega_R}([\mathcal{T}]_{\Omega_R\cup\Omega_C} - \mathcal{T}_k)]_{I,J^{\complement},:}$	$O\left(\lvert\Omega_R\rvert - \lvert\Omega_U\rvert\right)$
Line 5: $[\mathcal{C}_{k+1}]_{I^{\complement},:,:} = [\mathcal{T}_k]_{I^{\complement},J,:} + \eta_C[P_{\Omega_C}([\mathcal{T}]_{\Omega_R\cup\Omega_C} - \mathcal{T}_k)]_{I^{\complement},J,:}$	$O\left(\lvert\Omega_C\rvert - \lvert\Omega_U\rvert\right)$
Line 6: $\mathcal{U}_{k+1} = \mathcal{H}_r([\mathcal{T}_k]_{I,J,:} + \eta_U[\mathcal{P}_{\Omega_R\cup\Omega_C}([\mathcal{T}]_{\Omega_R\cup\Omega_C} - \mathcal{T}_k)]_{I,J,:})$	$O(\max\{\lvert I\rvert\lvert J\rvert rn_3, \lvert J\rvert\lvert I\rvert n_3\log(n_3)\})$
Line 8: Updating $\mathcal{T}_{k+1}$	$O(r\lvert I\rvert n_2n_3 + r\lvert J\rvert n_1n_3)$

2.5 Numerical Experiments

This section presents the performance of our t-CCS based ITCURC through numerical experiments on both synthetic and real-world data. The computations are performed on one of shared nodes of the Computing Cluster with a 64-bit Linux system (GLNXA64), featuring Intel(R) Xeon(R) Gold 6148 CPU (2.40 GHz). All experiments are carried out using MATLAB 2022a.

2.5.1 Synthetic data examples

This section evaluates ITCURC for t-CCS tensor completion, exploring the needed sample sizes and the impact of Bernoulli sampling probability and fiber sampling rates on low-tubal-rank tensor recovery.

We assess ITCURC's tensor recovery capability under different combinations of horizontal and lateral slice numbers $|I| = \delta n_1, |J| = \delta n_2$ and Bernoulli sampling rates p on selected subtensors. The study uses tensors of size $768 \times 768 \times 256$ with tubal-ranks $r \in \{2, 5, 7\}$. To counteract

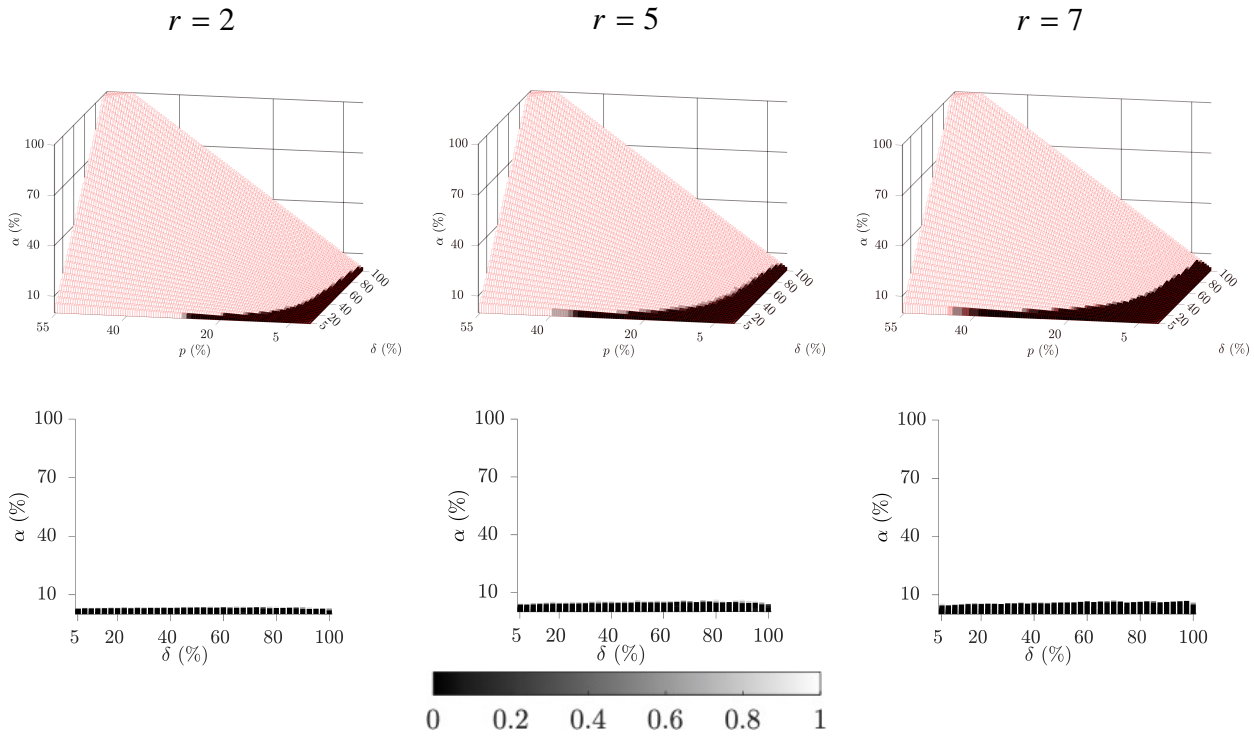


Figure 2.6 (**Row 1**) 3D and (**Row 2**) 2D views illustrate ITCURC's empirical phase transition for the t-CCS model. $\delta = |I|/768 = |J|/768$ shows sampled indices ratios, p is the Bernoulli sampling probability over subtensors, and α is the overall tensor sampling rate. White and black in the $768 \times 768 \times 256$ tensor results represent success and failure, respectively, across 25 tests for tubal ranks 2, 5, and 7 (Columns 1-3). The α needed for success remains consistent across different combinations δ and p.

randomness, we conduct 25 tests for each (δ, p, r) set, a test is successful if

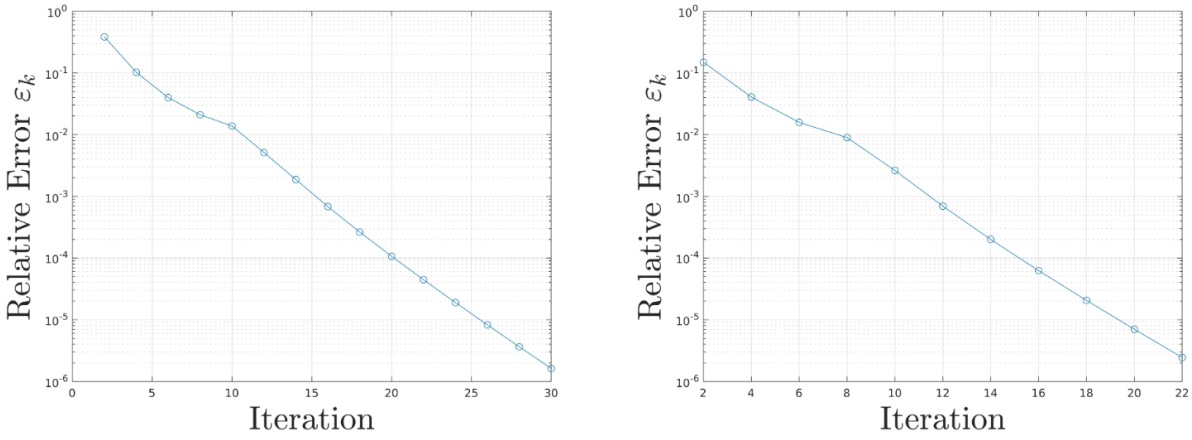
$$\varepsilon_k := \frac{\left\| \mathcal{T} - \mathcal{C}_k * \mathcal{W}_k^{\dagger} * \mathcal{R}_k \right\|_{\mathrm{F}}}{\|\mathcal{T}\|_{\mathrm{F}}} \leq 10^{-3}.$$

Our empirical phase transition results are presented in Figure 2.6, with the first row showing a 3D view of the phase transition results and the second row the corresponding 2D view. White and black pixels in these visuals indicate all tests' success and failure, respectively. The results highlight that higher overall sampling rates are needed for successful completion with larger tubal-ranks r. Importantly, tensor completion is achievable with sufficiently large overall sampling rates, regardless of the specific horizontal, lateral slice sizes, and subtensor sampling rates (see the results of 2D view). This demonstrates ITCURC's flexibility in sampling low-tubal-rank tensors for successful reconstruction. Additionally, we include our numerical results for the convergence of TICURC in the following section. In the following, we include further empirical data demonstrating the conver-

gence behavior of the ITCURC algorithm within the t-CCS model framework. In this experiment, we form a low tubal-rank tensor $\mathcal{T} = \mathcal{A} * \mathcal{B} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ using two Gaussian random tensors, where $\mathcal{A} \in \mathbb{R}^{n_1 \times r \times n_3}$ and $\mathcal{B} \in \mathbb{R}^{r \times n_2 \times n_3}$. Our objective is to examine the convergence behavior of the TICURC algorithm under different conditions. For the simulations, we set $n_1 = n_2 = 768$ and $n_3 = 256$, and generate partial observations using the t-CCS model by adjusting the rank r and configuring the concentrated subtensors as $\mathcal{R} \in \mathbb{R}^{\delta n_1 \times n_2 \times n_3}$ and $\mathcal{C} \in \mathbb{R}^{n_1 \times \delta n_2 \times n_3}$, with $0 < \delta < 1$. For each fixed r, we maintain a constant overall sampling rate α.

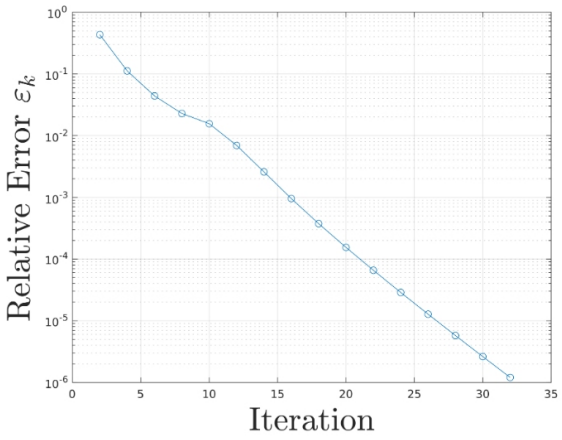
Utilizing the observed data, the TICURC algorithm is then employed to approximate the original low tubal-rank tensor. The algorithm continues until the stopping criterion $\varepsilon_k \leq 10^{-6}$ is met, where ε_k represents the relative error between the estimate at the k-th iteration and the actual tensor, defined as $\varepsilon_k = \frac{\|\mathcal{T} - \widehat{\mathcal{T}}_k\|_F}{\|\mathcal{T}\|_F}$.

For each specified set of parameters (r, δ, α), we generate 10 different tensor completion scenarios. The mean relative errors ε_k, along with the specific configurations, are reported in Figures 2.7 to 2.10. One can see that TICURC can achieve an almost linear convergence rate.

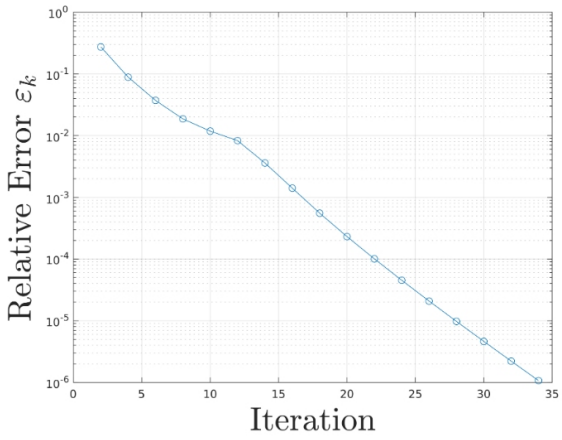


(a) $r = 2, \alpha = 0.15$ (b) $r = 5, \alpha = 0.25$

Figure 2.7 The averaged relative error of TICURC under the t-CCS model with respect to iterations over 10 independent trials with $\delta = 0.20$.

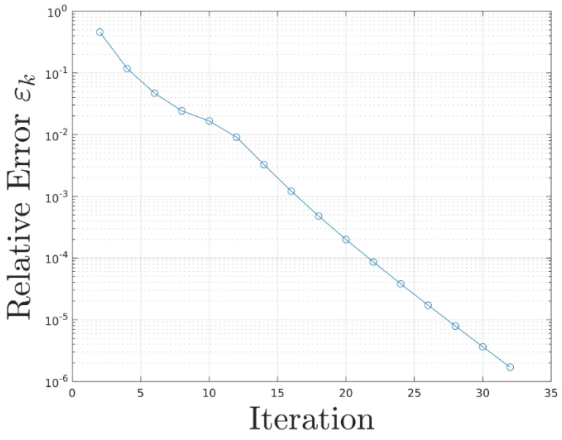


(a) $r = 2, \alpha = 0.15$.

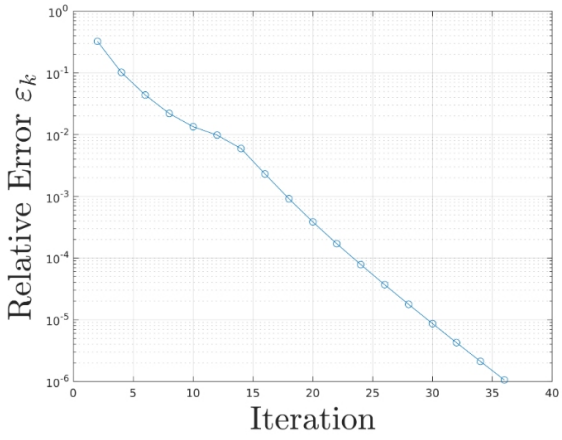


(b) $r = 5, \alpha = 0.25$

Figure 2.8 The averaged relative error of TICURC under the t-CCS model with respect to iterations over 10 independent trials with $\delta = 0.25$.



(a) $r = 2, \alpha = 0.15$

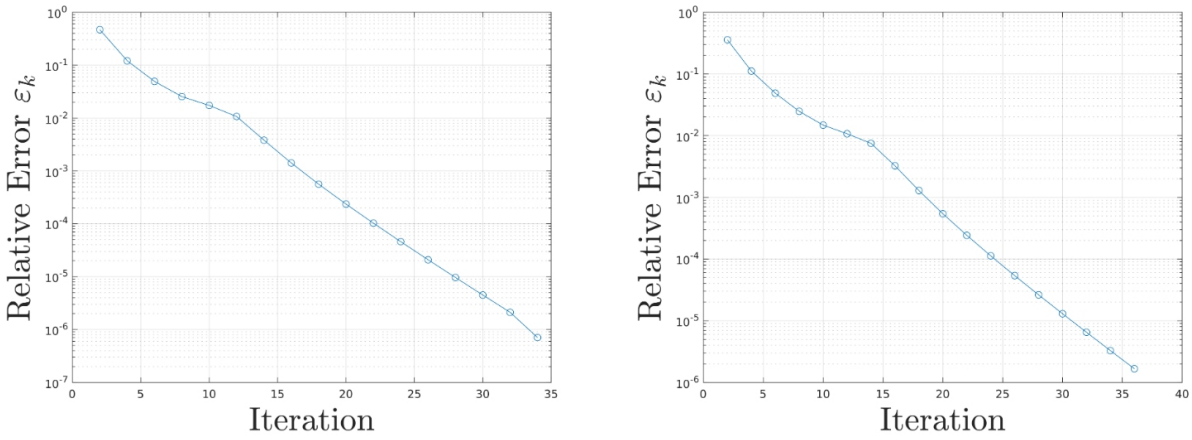


(b) $r = 5, \alpha = 0.25$

Figure 2.9 The averaged relative error of TICURC under the t-CCS model with respect to iterations over 10 independent trials with $\delta = 0.30$.

2.5.2 Real-world Applications

This section presents an evaluation and comparison between the t-CCS model and the Bernoulli Sampling model through tensor completion tasks across various types of data. Our goal is to examine assess the practical feasibility and real-world applicability of the t-CCS model, emphasizing its effectiveness in diverse operational environments. Our experiments are designed to compare the performance of ITCURC, designed based on t-CCS model, against with established TC methods designed based on Bernoulli sampling model such as BCPF [140], TMac [129], TNN [138], and



(a) $r = 2, \alpha = 0.15$ (b) $r = 5, \alpha = 0.25$

Figure 2.10 The averaged relative error of TICURC under the t-CCS model with respect to iterations over 10 independent trials with $\delta = 0.35$.

F-TNN [66]. Our test metric focuses on the quality and execution time of the reconstruction. Quality is assessed using the Peak Signal-to-Noise Ratio (PSNR) and the Structural Similarity Index (SSIM) where

$$\text{PSNR} = 10 \log_{10} \left(\frac{n_1 n_2 n_3 \|\mathcal{T}\|_\infty^2}{\|\mathcal{T} - \widetilde{\mathcal{T}}\|_{\text{F}}^2} \right).$$

and SSIM evaluates the structural similarity between two images, as detailed in [125]. On account of that the data are third-order tensors, we report the mean values of SSIM of all the frontal slices. Higher PSNR and SSIM scores suggest better reconstruction quality.

Our experimental process is as follows. We first generate random observations via the t-CCS model: uniformly randomly selecting concentrated horizontal ($\mathcal{R}$) and lateral (C) subtensors, defined as $\mathcal{R} = [\mathcal{T}]_{I,:,:}$ and $C = [\mathcal{T}]_{:,J,:}$, with $|I| = \delta n_1$ and $|J| = \delta n_2$; entries in $\mathcal{R}$ and C are sampled based on the Bernoulli sampling model with locations of the observed entries denoted by $\Omega_{\mathcal{R}}$ and Ω_C. The procedure of the t-CCS model results in a tensor that's only partially observed, primarily in the $\mathcal{R}$ and C. ITCURC is then applied to estimate the missing entries and thus recover the original tensor. For comparison, we also generate observations of the entire original tensor $\mathcal{T}$ using the Bernoulli sampling model with a probability $p_{\mathcal{T}} := \frac{|\Omega_{\mathcal{R}} \cup \Omega_C|}{n_1 n_2 n_3}$. Additionally, we estimate the missing data using several tensor completion methods: BCPF[1], which is based on the CP decomposition

[1] https://github.com/qbzhao/BCPF

framework; TMac[2], which utilizes the Tucker decomposition framework; and TNN[3] and F-TNN[4], which are both grounded in the t-SVD framework. To ensure reliable results, we repeat this entire procedure 30 times, averaging the PSNR and SSIM scores and the runtime to minimize the effects of randomness.

2.5.2.1 Color image completion

Color images, viewed as 3D tensors with dimensions for height, width, and color channels, are effectively modeled as low-tubal-rank tensors [80, 83]. In our tests, we focus on two large-size images: 'Building'[5] (of size $2579 \times 3887 \times 3$) and 'Window'[6] (of size $3009 \times 4513 \times 3$). We present averaged test results over various overall observation rates (α) in Table 2.2, and visual comparisons at $\alpha = 20\%$ in Figure 2.11.

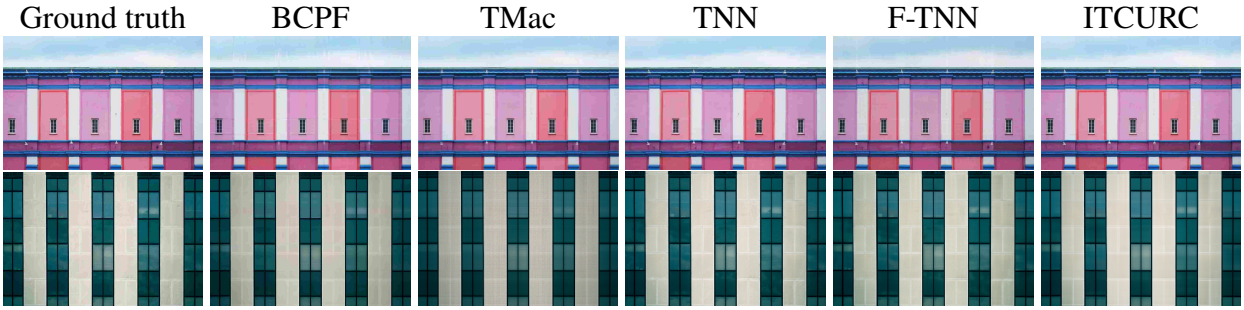


Figure 2.11 The visualization of color image inpainting for Building and Window datasets by setting tubal-rank $r = 35$ with the percentage selected horizontal and lateral slices $\delta = 13\%$ with overall sampling rate 20% for TICUR algorithm, while other algorithms are applied based on Bernoulli sampling model with the same overall sampling rate 20%. Additionally, t-CCS samples on the Building for ITCURC are the same as those in Figure 2.5.

Figure 2.11 presents a clear visual comparison of results from different methods at a 20% overall sampling rate, where the algorithms BCPF, TMac, TNN, F-TNN are applied on the Bernoulli Sampling model and ITCURC are applied on t-CCS model. Ground truth is the original image of a building, and a window. BCPF underperforms in visual effects compared to other methods. TNN shows slight variations from ground truth, maintaining colors and details with minor discrepancies.

[2]https://xu-yangyang.github.io/TMac/
[3]https://github.com/jamiezeminzhang/
[4]https://github.com/TaiXiangJiang/Framelet-TNN
[5]https://pxhere.com/en/photo/57707
[6]https://pxhere.com/en/photo/1421981

TMac reveals some notable differences. F-TNN improves reflection fidelity and color saturation, closely resembling ground truth. ITCURC also achieves the high similarity to Ground truth, accurately reproducing colors and details. Moreover, ITCURC significantly outperforms TMac, TNN, and F-TNN in the t-CCS based color image completion task, evidenced by the unsatisfactory results of BCPF, TMac, TNN, and F-TNN under t-CCS model, as illustrated in Figure 2.5.

Table 2.2 Image inpainting results on the Building and Window datasets. The **best results** are emphasized in bold, while the second-best results are underlined. ITCURC-δ refers to the ITCURC method with the percentages of selected horizontal and lateral slices set at a fixed rate of δ%. The t-CCS based algorithm ITCURC-δ%s are performed on t-CCS scheme while other Bernoulli based algorithms are performed on Bernoulli Sampling scheme.

Dataset		Building			Window		
Overall Observation Rate		12%	16%	20%	12%	16%	20%
PSNR	ITCURC-11	**28.9249**	**31.0050**	32.1645	35.2830	36.1611	37.0236
	ITCURC-12	28.5518	30.8055	31.9489	35.1195	36.1145	37.0174
	ITCURC-13	28.1893	30.7260	31.6825	35.0196	36.1215	36.8885
	BCPF	26.7939	28.2949	29.4298	30.1611	33.9990	35.4780
	TMac	27.0425	30.1755	32.3632	33.2673	36.6370	**37.5877**
	TNN	26.3466	30.3844	31.7512	31.8747	34.6443	36.7893
	F-TNN	28.2529	30.1521	**33.1660**	**35.6747**	**36.9233**	37.2618
SSIM	ITCURC-11	0.8310	**0.8880**	**0.9118**	0.8571	0.8738	0.8848
	ITCURC-12	0.8172	0.8818	0.9033	0.8535	0.8733	0.8850
	ITCURC-13	0.8016	0.8774	0.8954	0.8504	0.8731	0.8837
	BCPF	**0.8639**	0.8761	0.8873	0.8269	0.8554	0.8727
	TMac	0.8402	0.8586	0.9111	0.8200	**0.8928**	0.9035
	TNN	0.6458	0.8257	0.8382	0.8333	0.8564	0.8804
	F-TNN	0.7583	0.8354	0.8626	**0.8745**	0.8899	**0.9066**
Runtime (sec)	ITCURC-11	10.9354	**17.3187**	**18.1098**	**23.8990**	**24.1286**	**25.1853**
	ITCURC-12	**10.7715**	19.3517	19.7731	25.5856	26.3275	28.0392
	ITCURC-13	12.2208	21.2458	22.0287	28.8653	29.4986	30.8361
	BCPF	213.6800	360.2903	613.3072	345.3425	500.3060	1629.8061
	TMac	92.9568	104.8518	108.6827	233.8853	242.7499	259.6068
	TNN	3651.4556	3289.5535	3004.6557	5801.1631	6572.9697	6690.7945
	F-TNN	2642.9409	2692.6197	2267.5622	4739.2703	4134.5206	4105.0327

Table 2.2 shows ITCURC typically offers quality that is comparable to that of Bernoulli Sampling based TC algorithms. In runtime efficiency, ITCURC leveraging the t-CCS model significantly surpasses BCPF, TMac, TNN, and F-TNN, all of which are based on the Bernoulli sampling model. This efficiency enhancement highlights the t-CCS model's superior performance in practical applications. Additionally, ITCURC's consistent performance in delivering similar quality results across different δ, provided the overall sampling rates are consistent. These highlight the

flexibility and feasibility of the t-CCS model.

2.5.2.2 MRI reconstruction

In this study, we test on a MRI heart dataset[7] (of size $320 \times 320 \times 110$), where compact t-SVD with tubal-rank 35 yields less than 10% error, suggesting low-tubal-rank property of dataset. The visualization of reconstruction of MRI data using different methods at a 30% overall sampling rate are presented in Figure 2.12, and reconstruction quality and runtime are detailed in Table 2.3.

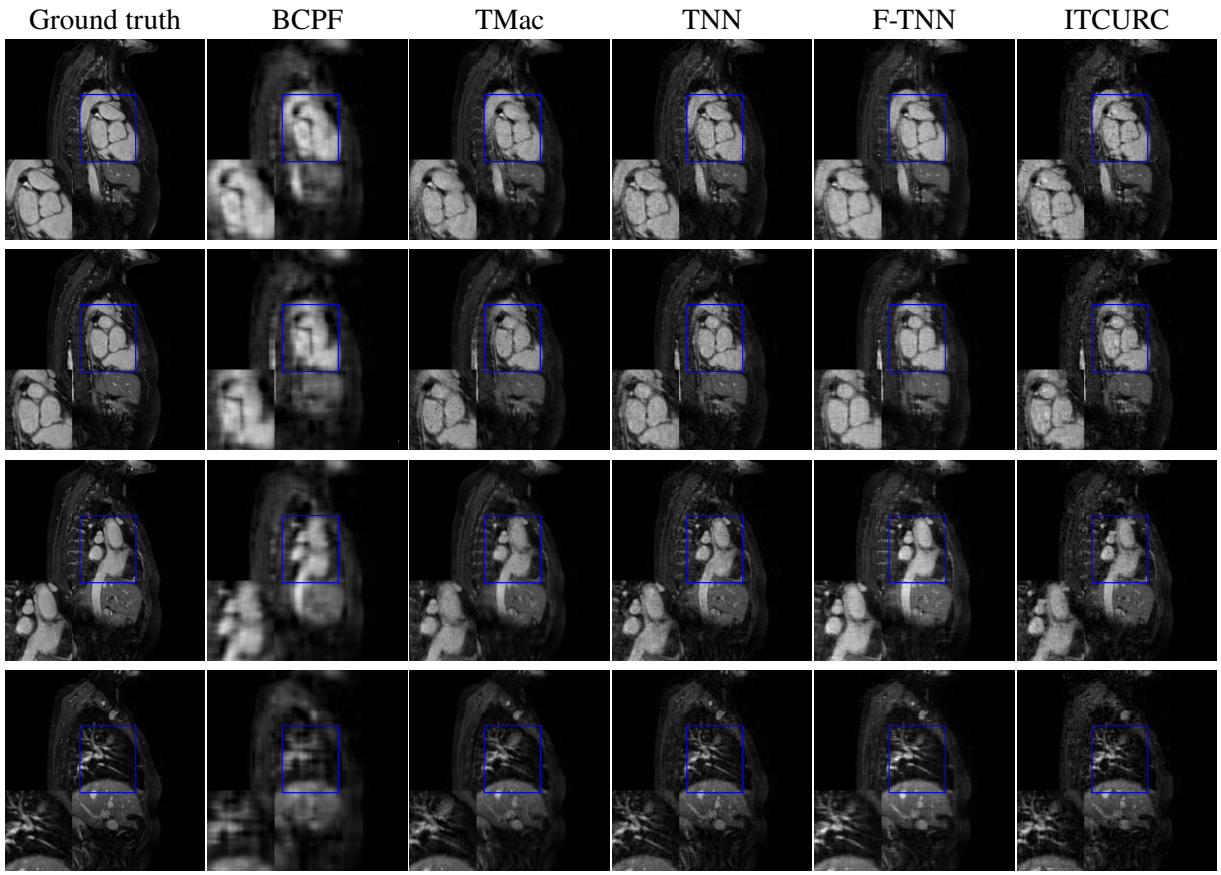


Figure 2.12 Visualizations of MRI data recovery using ITCURC with tubal rank $r = 35$, lateral and horizontal slice selection rate $\delta = 27\%$, and an overall sampling rate of 30%. Other algorithms are applied under Bernoulli sampling with the same overall sampling rate. Results for slices 51, 66, 86, and 106 are shown in rows 1 to 4, with a 1.3× magnified area at the bottom left of each result for clearer comparison.

Figure 2.12 shows recovery results for four frontal MRI slices using BCPF, TMac, TNN, F-TNN all under Bernoulli sampling model, and ITCURC under t-CCS model. The groundtruth serves as

[7] http://medicaldecathlon.com/dataaws

the actual dataset, from which missing values are to be predicted by the different algorithms. BCPF shows notable artifacts and lacks the sharp edges of the heart's interior structures. TMac improves over BCPF but still presents a softer representation of cardiac anatomy. TNN enhances the detail prediction, resulting in a more accurate completion of the tensor that begins to resemble the reference more closely. F-TNN maintains improvements on detail prediction, and edges within the cardiac structure suggest a refined approach to tensor completion. ITCURC shows a reconstruction where the cardiac structures are clearly defined, reflecting the structure present in the Ground truth without implying superiority, but rather indicating effectiveness in predicting the missing values. The highlighted regions of interest (ROIs), marked in blue, allow for a detailed comparison across the methods. In these regions of interest (ROIs), though ITCURC's reconstructions may not provide the most visually appealing results, they demonstrate efficiency in preserving structural integrity and texture, which are crucial aspects for clinical applications. Table 2.3 effectively demonstrates the flexibility and feasibility of the t-CCS model and we can see that reconstruction performance of t-CCS based method ITCURC generally aligns with, or matches, the reconstruction quality of Bernoulli-sampling-based TC methods. Furthermore, in terms of runtime efficiency, ITCURC, implemented under the t-CCS model, demonstrates a marked superiority by significantly outperforming alternatives such as BCPF, TMac, TNN, and F-TNN, all of which are applied under Bernoulli sampling scheme. This notable advantage distinctly underscores the enhanced effectiveness of the t-CCS model in practical applications.

2.5.2.3 Seismic data reconstruction

Geophysical 3D seismic data is often modeled as a tensor with inline, crossline, and depth dimensions. In our analysis, we focus on a seismic dataset[8] of size $51 \times 191 \times 146$, where compact t-SVD with tubal-rank 3 yields less than 5% error, suggesting low-tubal-rank property of dataset. The corresponding results are detailed in Figure 2.13 and Table 2.4.

Figure 2.13 presents the comparative analysis of seismic completion algorithms: BCPF, TMac, TNN, and F-TNN, applied based on the Bernoulli sampling model, in contrast to ITCURC, which is

[8]https://terranubis.com/datainfo/F3-Demo-2020

Table 2.3 The quantitative results for MRI data completion are presented, with **the best results** in bold and the second-best underlined. ITCURC-δ represents the ITCURC method specifying that the selected proportion of horizontal and lateral slices is exactly δ%. The t-CCS based algorithm ITCURC-δ%s are performed on t-CCS scheme while other Bernoulli based algorithms are performed on Bernoulli Sampling scheme.

Overall Observation Rate		10%	15%	20%	25%	30%
PSNR	ITCURC-23	22.4004	24.3553	26.9104	29.1861	30.3911
	ITCURC-25	22.2548	24.0435	26.9940	29.0219	31.1752
	ITCURC-27	22.1617	23.9311	27.0871	29.0699	31.2539
	BCPF	22.6581	24.5373	25.1663	25.8111	26.2042
	TMac	22.8690	25.4225	27.7802	29.1526	31.1648
	TNN	**23.4779**	25.3480	**27.9423**	28.4522	30.5580
	F-TNN	21.8172	**25.7453**	27.1969	**29.3630**	**31.3651**
SSIM	ITCURC-23	0.6020	0.6821	0.7584	0.8160	0.8451
	ITCURC-25	0.5990	0.6769	0.7571	0.8084	0.8619
	ITCURC-27	0.5990	0.6751	0.7567	0.8086	0.8600
	BCPF	**0.6817**	0.7151	0.7192	0.7301	0.7367
	TMac	0.6804	0.7323	0.7873	0.8227	**0.8924**
	TNN	0.6304	0.7494	0.7677	0.7984	0.8793
	F-TNN	0.6442	**0.7507**	**0.8181**	**0.8562**	0.8871
Runtime (sec)	ITCURC-23	5.7908	**7.0230**	**4.8030**	**5.3058**	**5.9484**
	ITCURC-25	**5.4241**	8.3488	5.5303	5.9375	6.7111
	ITCURC-27	5.8075	8.8408	6.0685	6.4371	7.4916
	BCPF	53.1651	88.2777	111.1949	180.6596	279.2789
	TMac	30.4813	28.0944	28.6216	28.9400	30.0219
	TNN	87.7591	84.0952	56.9761	57.9823	58.2098
	F-TNN	91.2048	86.3112	84.0228	82.2064	81.2119

applied based on the t-CCS model. The ground truth serves as the definitive reference, with its stark textural definition. BCPF falls short of delivering optimal fidelity, with finer details lost in translation. TMac is commendable for preserving the texture's integrity, providing a cohesive image. TNN improves upon this, sharpening textural nuances and closing in on the ground truth's visual quality. F-TNN excels visually, capturing essential texture information effectively, a significant advantage when the emphasis is on recognizing general features. ITCURC demonstrates comparable visual results though less effective than other methods in terms of PSNR and SSIM.

Table 2.4 shows that the t-CCS based method, ITCURC, achieves the fastest processing speeds while preserving satisfactory levels of PSNR and SSIM. This underscores the suitability of the t-CCS model for applications where rapid processing is essential without significant loss in visual

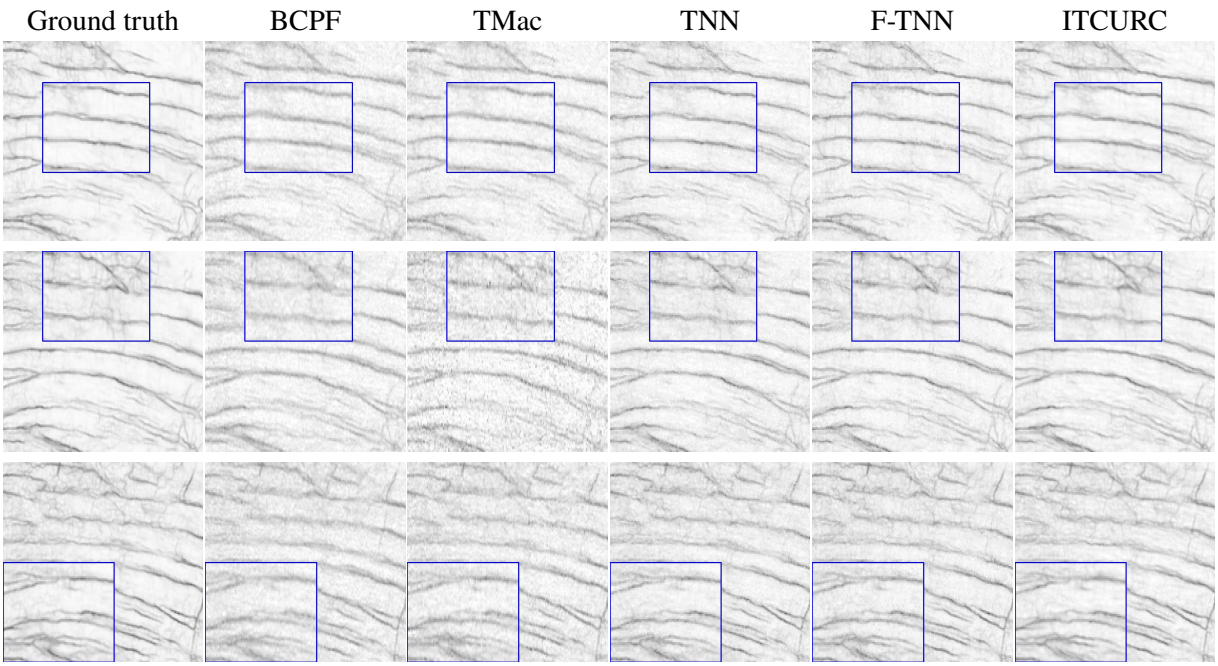


Figure 2.13 Visualization of seismic data recovery results by setting tubal-rank $r = 3$ for ITCURC with percentage of selected horizontal and lateral slices $\delta = 17\%$ with overall sampling rate 28% while other methods are applied based on Bernoulli sampling models with the same overall sampling rate 28%. Displayed are slices 15, 25, and 35 from top to bottom, with a 1.2× magnified area in each set for clearer comparison.

accuracy. Furthermore, the consistent performance of ITCURC across various subtensor sizes and sampling rates further emphasizes flexibility and feasibility of the t-CCS model in diverse operational environments.

Discussions on the results of real-world datasets

From the above results, it is evident that our method surpasses others in runtime with significantly lower computational costs. Consider a tensor of dimensions $n_1 \times n_2 \times n_3$. When a framelet transform matrix is constructed using n filters and l levels, the computational cost per iteration for framelet-based Tensor Nuclear Norm (F-TNN) is given by $O((nl-l+1)n_1n_2n_3(n_3+\min(n_1,n_2)))$. This formulation incorporates the processes involved in generating a framelet transform matrix, as elaborated in seminal works such as [21] and [65]. While enhancing the number of levels and filters in F-TNN can improve the quality of results, it also escalates the computational burden, particularly for tensors of substantial size. In our experiments, we have set both the framelet level and the number of filters to 1 for the F-TNN implementation. For comparison, the com-

Table 2.4 Quantitative results for seismic data completion: TMac, TNN, F-TNN with Bernoulli sampling, and our method with t-CCS. **Best results** are in bold, and second-best are underlined. ITCURC-δ refers to the ITCURC method with the percentages of selected horizontal and lateral slices set at a fixed rate of δ%. The t-CCS based algorithm ITCURC-δ%s are performed on t-CCS scheme while other Bernoulli based algorithms are performed on Bernoulli Sampling scheme.

Overall Observation Rate		12 %	16 %	20 %	24 %	28 %
PSNR	ITCURC-15	24.8020	26.4092	27.4143	29.3053	30.6585
	ITCURC-16	24.7386	26.1054	27.4737	29.3542	30.6905
	ITCURC-17	24.8381	26.1176	27.4768	28.8953	30.5312
	BCPF	24.0733	24.1905	24.2084	24.2454	24.3015
	TMac	**24.8859**	26.5349	26.9970	28.4662	30.7237
	TNN	23.7395	26.3806	27.7428	29.5430	30.9172
	F-TNN	24.0688	**27.5890**	**28.6408**	**29.7987**	**31.2791**
SSIM	ITCURC-15	0.5732	0.6691	0.7338	0.8143	0.8596
	ITCURC-16	0.5691	0.6507	0.7349	0.8129	0.8610
	ITCURC-17	0.5724	0.6491	0.7321	0.7939	0.8523
	BCPF	0.5304	0.5407	0.5420	0.5494	0.5532
	TMac	0.5566	0.6738	0.6962	0.7612	0.8504
	TNN	0.5165	0.6442	0.7577	0.8080	0.8486
	F-TNN	**0.6607**	**0.7551**	**0.8142**	**0.8479**	**0.8814**
Runtime (sec)	ITCURC-15	6.4327	6.7701	**6.2598**	**6.8633**	**6.8212**
	ITCURC-16	**6.3825**	**6.3579**	6.7522	7.0789	7.0215
	ITCURC-17	7.0379	6.6306	6.8325	7.1480	7.3253
	BCPF	33.5759	33.1832	32.1258	31.7875	31.2663
	TMac	16.6135	14.3412	16.8581	13.7124	13.1142
	TNN	34.5718	31.3138	29.2464	26.1727	23.9876
	F-TNN	22.1019	21.4482	22.1420	17.8848	18.0547

putational cost per iteration for the TNN is $O(\min(n_1, n_2)n_1n_2n_3 + n_1n_2n_3\log(n_3))$, and for the TMac, it is $O((r_1 + r_2 + r_3)n_1n_2n_3)$ where (r_1, r_2, r_3) denotes the Tucker rank. As for BCPF, it is $O(R^3(n_1n_2n_3) + R^2(n_1n_2 + n_2n_3 + n_3n_1))$, where R is the CP rank. In contrast, the computational expense per iteration of our proposed method is significantly reduced to $O(r|I|n_2n_3 + r|J|n_1n_3)$, assuming $|I| \ll n_1$ and $|J| \ll n_2$, indicating a substantial efficiency improvement over traditional methods.

Note that for F-TNN, [66] have formulated the tensor nuclear norm utilizing the M-product [70], a generalization of the t-product for 3-order tensor. In [66], they have incorporated a tight wavelet frame (framelet) as the transformation matrix M. This meticulous design of the M transformation contributes to the superior reconstruction quality of F-TNN. However, the absence of a rapid imple-

mentation for multiplying the tensor with matrix M along the third mode leads to F-TNN requiring significantly more computational time compared to other evaluated methods.

It is worth noting that our current approach provides an effective balance between runtime efficiency and reconstruction quality, making it well-suited for potential real-world applications. This balanced approach is particularly relevant in practical settings where it is essential to consider both speed and quality in big data applications.

2.6 Proofs of Theoretical Results

2.6.1 Proof of Theorem 2.4

In this section, we provide a detailed proof of Theorem 2.4, which is one of two important supporting theorems to our main result Theorem 2.6.

Before proceeding to prove Theorem 2.4, we will first introduce and discuss several supporting lemmas. These lemmas are crucial to establish the foundation for the proof of Theorem 2.4.

Lemma 2.2. *Let $\mathcal{T} \in \mathbb{K}^{n_1 \times n_2 \times n_3}$, $I \subseteq [n_1]$ and $J \subseteq [n_2]$. $\mathcal{S}_I$ and $\mathcal{S}_J$ are the horizontal and lateral sampling tensors associated with indices I and J respectively (see Definition 12). Then the following results hold*

$$\overline{\mathcal{S}_I * \mathcal{T}} = \begin{bmatrix} [\mathcal{S}_I]_{:,:,1} \cdot [\widehat{\mathcal{T}}]_{:,:,1} & & & \\ & [\mathcal{S}_I]_{:,:,1} \cdot [\widehat{\mathcal{T}}]_{:,:,2} & & \\ & & \ddots & \\ & & & [\mathcal{S}_I]_{:,:,1} \cdot [\widehat{\mathcal{T}}]_{:,:,n_3} \end{bmatrix}, \tag{2.15}$$

$$\overline{\mathcal{T} * \mathcal{S}_J} = \begin{bmatrix} [\widehat{\mathcal{T}}]_{:,:,1} \cdot [\mathcal{S}_J]_{:,:,1} & & & \\ & [\widehat{\mathcal{T}}]_{:,:,2} \cdot [\mathcal{S}_J]_{:,:,1} & & \\ & & \ddots & \\ & & & [\widehat{\mathcal{T}}]_{:,:,n_3} \cdot [\mathcal{S}_J]_{:,:,1} \end{bmatrix}. \tag{2.16}$$

Proof. Here, we will only focus on the proof of (2.15). First, it is easy to see that $\overline{\mathcal{S}_I * \mathcal{T}} = \overline{\mathcal{S}_I} \cdot \overline{\mathcal{T}}$.

In addition,

$$\overline{\mathcal{S}_I} = \begin{bmatrix} [\mathcal{S}_I]_{:,:,1} & & & \\ & [\mathcal{S}_I]_{:,:,1} & & \\ & & \ddots & \\ & & & [\mathcal{S}_I]_{:,:,1} \end{bmatrix}$$

and

$$\overline{\mathcal{T}} = \begin{bmatrix} [\widehat{\mathcal{T}}]_{:,:,1} & & & \\ & [\widehat{\mathcal{T}}]_{:,:,2} & & \\ & & \ddots & \\ & & & [\widehat{\mathcal{T}}]_{:,:,n_3} \end{bmatrix}.$$

The result can thus be derived. □

Theorem 2.7 ([115, 116])**.** *Consider a finite sequence $\{X_k\}$ of independent, random, Hermitian matrices with common dimension d. Assume that*

$$0 \le \lambda_{\min}(X_k) \text{ and } \lambda_{\max}(X_k) \le L \text{ for each index } k.$$

Set $Y = \sum_k X_k$. Let $\mu_{\min}$ and $\mu_{\max}$ be the minimum and maximum eigenvalues of $\mathbb{E}(Y)$ respectively. Then,

$$\mathbb{P}\{\lambda_{\min}(Y) \le (1-\varepsilon)\mu_{\min}\} \le d\left[\frac{\mathrm{e}^{-\varepsilon}}{(1-\varepsilon)^{1-\varepsilon}}\right]^{\mu_{\min}/L} \quad \text{for } \varepsilon \in [0,1), \text{ and}$$
$$\mathbb{P}\{\lambda_{\max}(Y) \ge (1+\varepsilon)\mu_{\max}\} \le d\left[\frac{\mathrm{e}^{\varepsilon}}{(1+\varepsilon)^{1+\varepsilon}}\right]^{\mu_{\max}/L} \quad \text{for } \varepsilon \ge 0.$$

Lemma 2.3. *Suppose A is a block diagonal matrix, i.e. $A = \begin{bmatrix} A_1 & & & \\ & A_2 & & \\ & & \ddots & \\ & & & A_{n_3} \end{bmatrix}$, where each A_i is a matrix of size $n_1 \times r_i$, $A_i^\top A_i = \mathbb{I}_{r_i}$ and $r_i \le n_1$ for $\forall i \in [n_3]$. Let I be a random subset of $[n_1]$.*

Then for any $\delta \in [0,1)$, the $\sum_{i=1}^{n_3} r_i$-th singular value of the matrix

$$N =: \begin{bmatrix} [\mathcal{S}_I]_{:,:,1} & & & \\ & [\mathcal{S}_I]_{:,:,1} & & \\ & & \ddots & \\ & & & [\mathcal{S}_I]_{:,:,1} \end{bmatrix} \begin{bmatrix} A_1 & & & \\ & A_2 & & \\ & & \ddots & \\ & & & A_{n_3} \end{bmatrix}$$

will be no less than $\sqrt{\frac{(1-\delta)|I|}{n_1}}$ with probability at least

$$1 - \|\vec{r}\|_1 e^{-(\delta+(1-\delta)\log(1-\delta))\frac{|I|}{n_1 \max_{i\in[n_1 n_3]} \|[A]_{i,:}\|_{\mathrm{F}}^2}}.$$

Proof. Firstly, it is easy to check that

$$N = \left[\mathbb{I} \otimes (\mathcal{S}_I)_{:,:,1}\right] \cdot A = \sum_{i\in I} \sum_{j=1}^{n_3} \mathbf{e}_{(j-1)n_1+i} \cdot [A]_{(j-1)n_1+i,:},$$

where $\mathbf{e}_{(j-1)n_1+i}$ is the standard column basis vector of $\mathbb{K}^{n_1 n_3}$. Consider $\sum_i^{n_3} r_i \times \sum_i^{n_3} r_i$ Gram matrix

$$\begin{aligned} Y := N^\top \cdot N &= \sum_{i\in I} \sum_{j=1}^{n_3} (\mathbf{e}_{(j-1)n_1+i} \cdot [A]_{(j-1)n_1+i,:})^\top \cdot \mathbf{e}_{(j-1)n_1+i} \cdot [A]_{(j-1)n_1+i,:} \\ &= \sum_{i\in I} \sum_{j=1}^{n_3} [A]_{(j-1)n_1+i,:}^\top [A]_{(j-1)n_1+i,:} =: \sum_{i\in I} T_i, \end{aligned}$$

where $T_i = \sum_{j=1}^{n_3} [A]_{(j-1)n_1+i,:}^\top [A]_{(j-1)n_1+i,:}$. It is easy to see that Y is a random matrix due to randomness inherited from the random set I. It is easy to see that each T_i is a positive semidefinite matrix of size $\sum_i^{n_3} r_i \times \sum_i^{n_3} r_i$. Thus, the random matrix Y in fact is a sum of $|I|$ random matrices sampled without replacement from the set $\{X_1, X_2, \cdots X_{n_1}\}$ of positive semi-definite matrices. Notice that

$$\begin{aligned} \lambda_{\max}(T_i) &= \lambda_{\max}\left(\sum_{j=1}^{n_3} (\mathbf{e}_{(j-1)n_1+i} \cdot [A]_{(j-1)n_1+i,:})^\top \cdot \mathbf{e}_{(j-1)n_1+i} \cdot [A]_{(j-1)n_1+i,:}\right) \\ &= \left|\sigma_{\max}\left(\sum_{j=1}^{n_3} \mathbf{e}_{(j-1)n_1+i} \cdot [A]_{(j-1)n_1+i,:}\right)\right|^2 \le \max_i \|[A]_{i,:}\|_{\mathrm{F}}^2. \end{aligned}$$

By the orthogonal property of matrix A, it is easy to see that $\mathbb{E}(T_i) = \frac{1}{n_1}\mathbb{I}$ and thus $\mathbb{E}(Y) = \frac{|I|}{n_1}$, where $\mathbb{E}$ is the expectation operator. Thus, by the fact that $\lambda_{\min}(Y) = \sigma_{\min}^2(N)$ and by the Chernoff

inequality (see Theorem 2.7), we have

$$\mathbb{P}\left(\sigma_{\min}(N) \leq \sqrt{\frac{(1-\delta)|I|}{n_1}}\right) \leq \|\vec{r}\|_1 e^{-(\delta+(1-\delta)\log(1-\delta))\frac{|I|}{n_1 \max\limits_{i\in[n_1 n_3]}\|[A]_{i,:}\|_{\mathrm{F}}^2}}, \forall \delta \in [0,1).$$

□

In the following, we delve into the proof of Theorem 2.4 to tell about how likely t-CUR decomposition holds.

The proof of Theorem 2.4. According to Theorem 2.3, $\mathcal{T} = \mathcal{C} * \mathcal{U}^{\dagger} * \mathcal{R}$ is equivalent to $\mathrm{rank}_m(\mathcal{T}) = \mathrm{rank}_m(\mathcal{C}) = \mathrm{rank}_m(\mathcal{R})$. Therefore, it suffices to prove that $\mathrm{rank}_m(\mathcal{T}) = \mathrm{rank}_m(\mathcal{C}) = \mathrm{rank}_m(\mathcal{R})$ holds with probability at least $1 - \frac{1}{n_1^{\beta_1}} - \frac{1}{n_2^{\beta_2}}$ with the given conditions. Notice that

$$\begin{aligned}
\overline{\mathcal{T}} &= \begin{bmatrix} [\widehat{\mathcal{T}}]_{:,:,1} & & & \\ & [\widehat{\mathcal{T}}]_{:,:,2} & & \\ & & \ddots & \\ & & & [\widehat{\mathcal{T}}]_{:,:,n_3} \end{bmatrix} \\
&= \begin{bmatrix} W_1\Sigma_1 V_1^{\top} & & & \\ & W_2\Sigma_2 V_2^{\top} & & \\ & & \ddots & \\ & & & W_{n_3}\Sigma_{n_3} V_{n_3}^{\top} \end{bmatrix} \\
&= \begin{bmatrix} W_1 & & & \\ & W_2 & & \\ & & \ddots & \\ & & & W_{n_3} \end{bmatrix} \cdot \begin{bmatrix} \Sigma_1 & & & \\ & \Sigma_2 & & \\ & & \ddots & \\ & & & \Sigma_{n_3} \end{bmatrix} \cdot \begin{bmatrix} V_1^{\top} & & & \\ & V_2^{\top} & & \\ & & \ddots & \\ & & & V_{n_3}^{\top} \end{bmatrix} \\
&=: W \cdot \Sigma \cdot V^{\top},
\end{aligned} \tag{2.17}$$

where $W_i\Sigma_i V_i^{\top}$ in (2.17) is the compact SVD of $[\overline{\mathcal{T}}]_{:,:,i}$ for $i \in [n_3]$. And $\overline{\mathcal{R}} = \overline{\mathcal{S}_I} W \Sigma V^{\top}$. By the definition of tensor multi-rank, we have $W_i \in \mathbb{K}^{n_1 \times r_i}$, $\Sigma_i \in \mathbb{K}^{r_i \times r_i}$, $V_i \in \mathbb{K}^{n_2 \times r_i}$, $\mathrm{W} \in \mathbb{K}^{n_1 n_3 \times \|\vec{r}\|_1}$, $\Sigma \in \mathbb{K}^{\|\vec{r}\|_1 \times \|\vec{r}\|_1}$, and $V \in \mathbb{K}^{n_2 n_3 \times \|\vec{r}\|_1}$.

Consequently, demonstrating that $\mathrm{rank}(\overline{\mathcal{R}}) = \|\vec{r}\|_1$ suffices to ensure the condition that $\mathrm{rank}_m(\mathcal{T}) =$

$\text{rank}_m(\mathcal{R})$. Observe that Σ is a square matrix with full rank and V has full column rank. By the Sylvester rank inequality, $\text{rank}(\overline{\mathcal{R}}) = \|\vec{r}\|_1$ can be guaranteed by showing $\text{rank}(\overline{S}_I \cdot W) = \|\vec{r}\|_1$. By applying Lemma 2.3, we have that for all $\delta \in [0, 1)$,

$$\mathbb{P}\left(\sigma_{\|\vec{r}\|_1}(\overline{S}_I \cdot W) \leq \sqrt{(1-\delta)|I|/n_1}\right) \leq \|\vec{r}\|_1 e^{-(\delta+(1-\delta)\log(1-\delta))\frac{|I|}{\mu_0\|\vec{r}\|_\infty}}.$$

$|I| \geq \frac{\beta_1\mu_0\|r\|_\infty \log(n_1\|\vec{r}\|_1)}{\delta+(1-\delta)\log(1-\delta)}$ implies $\mathbb{P}\left(\sigma_{\|\vec{r}\|_1}(\overline{S}_I \cdot W) \leq \sqrt{(1-\delta)|I|/n_1}\right) \leq \frac{1}{{n_1}^{\beta_1}}$.

Note that

$$\mathbb{P}\left(\text{rank}(\overline{S}_I \cdot W) < \|\vec{r}\|_1\right) \leq \mathbb{P}\left(\sigma_{\|\vec{r}\|_1}(\overline{S}_I \cdot W) \leq \sqrt{\frac{(1-\delta)|I|}{n_1}}\right).$$

We thus have when $|I| \geq \frac{\beta_1\mu_0\|r\|_\infty \log(n_1\|\vec{r}\|_1)}{\delta+(1-\delta)\log(1-\delta)}$,

$$\begin{aligned}\mathbb{P}\left(\text{rank}(\overline{S}_I \cdot W) = \|\vec{r}\|_1\right) =& 1 - \mathbb{P}\left(\text{rank}(\overline{S}_I \cdot W) < \|\vec{r}\|_1\right) \\ \geq& 1 - \mathbb{P}\left(\sigma_{\|\vec{r}\|_1}(\overline{S}_I \cdot W) \leq \sqrt{\frac{(1-\delta)|I|}{n_1}}\right) \geq 1 - \frac{1}{{n_1}^{\beta_1}}.\end{aligned}$$

Similarly, one can show that $\text{rank}(\overline{\mathcal{S}}_J \cdot \overline{\mathcal{V}}) = \|\vec{r}\|_1$ holds with probability at least $1 - \frac{1}{{n_2}^{\beta_2}}$ provided that $|J| \geq \frac{\beta_2\mu_0\|r\|_\infty \log(n_2\|\vec{r}\|_1)}{\delta+(1-\delta)\log(1-\delta)}$.

Combining all the statements and setting $\delta = 0.815$ and $\beta_1 = \beta_2 = \beta$, we conclude that $\mathcal{T} = \mathcal{C} * \mathcal{U}^\dagger * \mathcal{R}$ holds with probability at least $1 - \frac{1}{n_1^\beta} - \frac{1}{n_2^\beta}$, provided that $|I| \geq 2\beta\mu_0\|\vec{r}\|_\infty \log(n_1\|\vec{r}\|_1)$ and $|J| \geq 2\beta\mu_0\|\vec{r}\|_\infty \log(n_2\|\vec{r}\|_1)$. □

2.6.1.1 Some remarks on the proof of Theorem 2.4

We wish to emphasize that the techniques employed in our proof are not merely straightforward extensions of the probabilistic estimates used in matrix CUR decompositions since one cannot directly apply the union of matrix CUR probabilistic estimates to flattened tensors due to ‘‘dependence’’ and ”intertwined” sampling property of each sub block matrix after one flattens a tensor to a block diagonal matrix in the Fourier domain. We introduce a new tool that offers a probabilistic estimate for achieving an exact t-CUR decomposition, utilizing a novel proof methodology. The cornerstone of our approach is to assess the likelihood that multi-rank is preserved when selecting horizontal or lateral slices uniformly. This approach distinguishes our method from traditional techniques applied in matrix settings. Although our method involves converting a third-order tensor

into a block diagonal matrix, it necessitates the introduction of innovative techniques. These are required to overcome several challenges that do not arise in matrix-based proofs. Given a three-order tensor $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ with multi-rank $\vec{r} = (r_1, r_2, \cdots, r_{n_3})$, its flattened version in the Fourier domain is denoted as

$$\overline{\mathcal{T}} = \begin{bmatrix} [\widehat{\mathcal{T}}]_{:,:,1} & 0 & 0 & \cdots & 0 \\ 0 & [\widehat{\mathcal{T}}]_{:,:,2} & 0 & \cdots & 0 \\ 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & [\widehat{\mathcal{T}}]_{:,:,n_3} \end{bmatrix},$$

where $\widehat{\mathcal{T}} = \mathrm{FFT}(\mathcal{T}, [], 3)$. For simplicity, we denote T_i as $[\widehat{\mathcal{T}}]_{:,:,i}$ for $i = 1, \cdots, n_3$. It is easy to see that sampling horizontal(lateral) slices of tensor $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ with an index set I is equivalent of sampling row(column) vectors of the matrix $\widehat{\mathcal{T}}$ with indexes $I, n_1 + I, \cdots, (n_3 - 1)n_1 + I$. In other words, the process of sampling I horizontal(lateral) slices is the same with sampling I rows(columns) of T_i, for $i = 1, \cdots, n_3$. Similar arguments for the lateral slice index set J.

Consider the sample space

$$\Omega = \{(I, J), I \subset \{1, \cdots, n_1\}, J \subset \{1, \cdots, n_2\}\}.$$

Define the events $\mathcal{F}_i$ as

$$\{(I, J) \subset \Omega \mid [T_i]_{I,J} = r_i\},$$

for $i = 1, \ldots, n_3$.

Lemma 2.4.

$$\{(I, J) : [T_i]_{I,J} = r_i, i = 1, \cdots, n_3\} \subsetneq \mathcal{F}_1 \cap \mathcal{F}_2 \cap \cdots \cap \mathcal{F}_{n_3}.$$

Proof. The set $\{(I, J) : [T_i]_{I,J} = r_i, i = 1, \cdots, n_3\}$ can be viewed as a product space, i.e.,

$$\{\underbrace{(I, J) \times (I, J) \cdots \times (I, J)}_{\text{n times}} : [T_i]_{I,J} = r_i, i = 1, \cdots, n_3\}.$$

However,

$$\mathcal{F}_1 \cap \mathcal{F}_2 \cap \cdots \mathcal{F}_{n_3} = \{(I_1, J_1) \times (I_2, J_2) \times \cdots (I_{n_3}, J_{n_3}) : (I_i, J_i) \in \mathcal{F}_i, i = 1, \cdots, n_3\}.$$

Therefore,

$$\{(I,J) : [T_i]_{I,J} = r_i, i = 1, \cdots, n_3\} \subsetneq \mathcal{F}_1 \cap \mathcal{F}_2 \cap \cdots \cap \mathcal{F}_{n_3}.$$

□

Let $\mathcal{G}$ represent the event

$$\{(I,J) \subset \Omega \mid |I| \geq \mu_0 |\vec{r}| \log(n_1) \log(r), |J| \geq \mu_0 |\vec{r}| \log(n_2) \log(r)\}.$$

Although one might have that the conditional probability inequality, based on the [32, Theorem 2.1],[19, Theorem 2],

$$\mathbb{P}(\mathcal{F}_i|\mathcal{G}) \geq 1 - \frac{4r^2}{n_1 n_2},$$

one can find that based on Lemma 2.4:

$$\begin{aligned}\mathbb{P}(\{(I,J) \subset \Omega : \text{rank}_m([\mathcal{T}]_{I,J,:}) = \vec{r}\}|\mathcal{G}) = \vec{r}|\mathcal{G}) = \mathbb{P}(\{(I,J) \subset \Omega : [T_i]_{I,J} = r_i, i = 1, \cdots, n_3\}|\mathcal{G}) \\ \leq \mathbb{P}(\mathcal{F}_1 \cap \mathcal{F}_2 \cdots \cap \mathcal{F}_{n_3}|\mathcal{G})\end{aligned}$$

The probability inequality $\mathbb{P}(\{(I,J) \subset \Omega : \text{rank}_m([\mathcal{T}]_{I,J,:}) = \vec{r}\}|\mathcal{G}) \leq \mathbb{P}(\mathcal{F}_1 \cap \mathcal{F}_2 \cdots \cap \mathcal{F}_{n_3}|\mathcal{G})$ directly stops us from applying $\mathbb{P}(\mathcal{F}_1 \cap \mathcal{F}_2 \cdots \cap \mathcal{F}_{n_3}|\mathcal{G}) \geq \sum_{i=1}^{n_3} \mathbb{P}(\mathcal{F}_i|\mathcal{G}) - (n_3 - 1)$. As a result, we can not get a lower bound of $\mathbb{P}(\{(I,J) \subset \Omega : [T_i]_{I,J} = r_i, i = 1, \cdots, n_3\}|\mathcal{G}) = \vec{r}|\mathcal{G})$ via a direct union of matrix CUR probabilistic estimate results.

Furthermore, we hope to emphasize that applying the matrix Chernoff inequality to a flatten tensor also presents numerous challenges. Notice that one flatten tensor $\mathcal{T}$ into $\overline{\mathcal{T}}$ in the Fourier domain, the corresponding sampling index set of rows with respect to $\overline{\mathcal{T}}$ becomes $\bigcup_{i \in I} I_i$, where

$$I_i := \{i, n_1 + i, 2n_1 + i, \cdots, (n_3 - 1)n_1 + i\}, i = 1, \cdots, n_1,$$

and the corresponding sampling index of columns with respect to $\overline{\mathcal{T}}$ becomes $\bigcup_{i \in I} J_i$, where

$$J_i := \{i, n_2 + i, 2n_2 + i, \cdots, (n_3 - 1)n_2 + i\}, i = 1, \cdots, n_2.$$

Without loss of generality and for the sake of brevity, in the following, we focus solely on the case of selecting horizontal slices of $\mathcal{T}$, denoted by the sampling index set I. It is easy to find that one can not directly apply the matrix Chernoff inequality to the finite set of positive-semi-definite matrices

$$\mathcal{H} = \left\{M_1, M_2, \cdots M_{n_3}, M_{n_3+1}, M_{n_3+2}, \cdots, M_{2n_3}, \cdots, M_{n_1 n_3}\right\},$$

where $M_j = [\overline{\mathcal{T}}]_{j,:}^\top \cdot [\overline{\mathcal{T}}]_{j,:}$ with $j = 1, \cdots, n_1 n_3$ due to the intertwined property of sampling index set $\bigcup_{i \in I} I_i$. Specifically, the intertwined nature of the index set $\bigcup_{i \in I} I_i$ complicates the estimation of the spectral bound for the sum of random matrices. Specifically, the expression $|\sigma_{\max}(\sum_{j=1}^{n_1} \mathbf{e}(j-1)n_3 + i \cdot [\overline{\mathcal{T}}]_{(j-1)n_3+i,:})|^2$ does not serve as a lower bound for $\max_i |[\overline{\mathcal{T}}]_{i,:}|_2^2$.

2.6.2 Proof of Theorem 2.5

In this section, we provides a detailed proof of Theorem 2.5, another important supporting theorem to our main theoretical result Theorem 2.6. To the best of our knowledge, there is no existing tensor version of the result found in [97, Theorem 1.1], which furnishes an explicit expression of numerical constants within the theorem's statement. Existing results related to tensor versions, such as [138, Theorem 3.1] in the context of tensor completion, typically only imply numerical constants implicitly. One can see that [138, Theorem 3.1] does not give an explicit expression of numerical constants of c_0, c_1 and c_2.

Theorem 2.8. *[138, Theorem 3.1] Suppose $\mathcal{M}$ is an $n_1 \times n_2 \times n_3$ tensor and its reduced t-SVD is given by $\mathcal{M} = \mathcal{U} * \mathcal{S} * \mathcal{V}^\top$ where $\mathcal{U} \in \mathbb{R}^{n_1 \times r \times n_3}$, $\mathcal{S} \in \mathbb{R}^{r \times r \times n_3}$, and $\mathcal{V} \in \mathbb{R}^{n_2 \times r \times n_3}$. Suppose $\mathcal{M}$ satisfies the standard tensor incoherent condition with parameter $\mu_0 > 0$. Then there exists constants $c_0, c_1, c_2 > 0$ such that if*

$$p \geq c_0 \frac{\mu_0 r \log(n_3(n_1+n_2))}{\min\{n_1, n_2\}}.$$

Then $\mathcal{M}$ is the unique minimizer to the follow optimization

$$\begin{aligned} \min_{\mathcal{X}} \quad & \|\mathcal{X}\|_{TNN} \\ \text{subject to} \quad & \mathcal{P}_\Omega(\mathcal{X}) = \mathcal{P}_\Omega(\mathcal{M}), \end{aligned}$$

with probability at least

$$1 - c_1((n_1+n_2)n_3)^{-c_2}.$$

Our work constitutes a substantial contribution through the meticulous analysis of these numerical constants, yielding explicit formulations for their expressions. The details of these theoretical advancements are comprehensively elaborated in our theoretical section. Before moving forward,

let us introduce several notations used throughout the rest of the supplemental material but not covered in earlier sections.

Definition 15. *Suppose $\mathcal{T}$ is an $n_1 \times n_2 \times n_3$ tensor and its compact t-SVD is given by $\mathcal{T} = \mathcal{U} * \mathcal{S} * \mathcal{V}^\top$ where $\mathcal{U} \in \mathbb{K}^{n_1 \times r \times n_3}, \mathcal{S} \in \mathbb{K}^{r \times r \times n_3}$ and $\mathcal{V} \in \mathbb{K}^{n_2 \times r \times n_3}$. Define projection space $\mathbb{T}$ as*

$$\left\{ \sum_{k=1}^{r} ([\mathcal{U}]_{:,k,:} * \mathcal{X}_k^\top + \mathcal{Y}_k * [\mathcal{V}]_{:,k,:}^\top) : \mathcal{X}_k \in \mathbb{K}^{n_2 \times 1 \times n_3}, \mathcal{Y}_k \in \mathbb{K}^{n_1 \times 1 \times n_3} \right\}$$

and the orthogonal projection space $\mathbb{T}^\perp$ is the orthogonal complement $\mathbb{T}$ in $\mathbb{K}^{n_1 \times n_2 \times n_3}$. Define $\mathcal{P}_\mathbb{T}(\mathcal{X})$ and $\mathcal{P}_{\mathbb{T}^\perp}(\mathcal{X})$ as

$$\mathcal{P}_\mathbb{T}(\mathcal{X}) = \mathcal{U} * \mathcal{U}^\top * \mathcal{X} + \mathcal{X} * \mathcal{V} * \mathcal{V}^\top - \mathcal{U} * \mathcal{U}^\top * \mathcal{X} * \mathcal{V} * \mathcal{V}^\top,$$

$$\mathcal{P}_{\mathbb{T}^\perp}(\mathcal{X}) = \left(\mathcal{I}_{n_1} - \mathcal{U} * \mathcal{U}^\top\right) * \mathcal{X} * \left(\mathcal{I}_{n_2} - \mathcal{V} * \mathcal{V}^\top\right),$$

where $\mathcal{I}_{n_1}$ is the identity tensor of size $n_1 \times n_1 \times n_3$ and $\mathcal{I}_{n_2}$ is the identity tensor of size $n_2 \times n_2 \times n_3$.

Definition 16. *Define the operator $\mathcal{R}_\Omega : \mathbb{K}^{n_1 \times n_2 \times n_3} \to \mathbb{K}^{n_1 \times n_2 \times n_3}$ as:*

$$\mathcal{R}_\Omega(\mathcal{X}) = \sum_{i,j,k} \frac{1}{p} \delta_{i,j,k} [\mathcal{X}]_{i,j,k} \mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^\top,$$

where $[\mathcal{X}]_{i,j,k}$ is the (i,j,k)-th entry of a tensor $\mathcal{X} \in \mathbb{K}^{n_1 \times n_2 \times n_3}$.

Definition 17. *Given two tensor $\mathcal{A} \in \mathbb{K}^{n_1 \times n_2 \times n_3}$ and $\mathcal{B} \in \mathbb{K}^{n_1 \times n_2 \times n_3}$, the inner product of these two tensors is defined as:*

$$\langle \mathcal{A}, \mathcal{B} \rangle = \frac{1}{n_3} \operatorname{trace}\left(\overline{\mathcal{B}}^\top \cdot \overline{\mathcal{A}}\right).$$

Before we introduce tensor operator norm, we need to introduce a transformed version of a tensor operator. Given a tensor operator, $\mathcal{F} : \mathbb{K}^{n_1 \times n_2 \times n_3} \to \mathbb{K}^{n_1 \times n_2 \times n_3}$, the associated transformed operator $\overline{\mathcal{F}} : \mathbb{B} \to \mathbb{B}$, where $\mathbb{B} = \left\{ \overline{\mathcal{B}} : \mathcal{B} \in \mathbb{K}^{n_1 \times n_2 \times n_3} \right\}$, is defined as

$$\overline{\mathcal{F}}(\overline{\mathcal{X}}) = \overline{\mathcal{F}(\mathcal{X})}.$$

Definition 18 (Tensor operator norm)**.** *Given a operator $\mathcal{F} : \mathbb{K}^{n_1 \times n_2 \times n_3} \to \mathbb{K}^{n_1 \times n_2 \times n_3}$, the operator norm $\|\mathcal{F}\|$ is defined as $\|\mathcal{F}\| = \|\overline{\mathcal{F}}\| = \max_{\|\overline{\mathcal{X}}\|_\mathrm{F}=1} \|\overline{\mathcal{F}}(\overline{\mathcal{X}})\|_\mathrm{F} = \max_{\|\overline{\mathcal{X}}\|_\mathrm{F}=1} \left\|\overline{\mathcal{F}(\mathcal{X})}\right\|_\mathrm{F}$.*

Definition 19 ($l_{\infty,2}$ norm [138])**.** *Given a tensor $\mathcal{X} \in \mathbb{K}^{n_1 \times n_2 \times n_3}$, its $l_{\infty,2}$ norm is defined as*

$$\|\mathcal{X}\|_{\infty,2} := \max\{\max_i \sqrt{\sum_{b,k} [\mathcal{X}]_{i,b,k}^2}, \max_j \sqrt{\sum_{a,k} [\mathcal{T}]_{a,j,k}^2}\}.$$

Definition 20 (Tensor infinity norm)**.** *Given a tensor* $\mathcal{X} \in \mathbb{K}^{n_1 \times n_2 \times n_3}$*, the tensor infinity norm of it is defined as* $\|\mathcal{X}\|_\infty := \max_{i,j,k} |[\mathcal{X}]_{i,j,k}|$.

In the following, we will present a formal definition of the tensor completion problem based on the Bernoulli sampling model. Consider a third-order tensor $\mathcal{T} \in \mathbb{K}^{n_1 \times n_2 \times n_3}$ with tubal-rank r. We denote Ω as the set of indices of the observed entries. Suppose that Ω is generated according to the Bernoulli sampling model with probability p. We define the sampling operator $\mathcal{P}_\Omega$ such that for a given tensor $\mathcal{X}$ in $\mathbb{K}^{n_1 \times n_2 \times n_3}$,

$$\mathcal{P}_\Omega(\mathcal{X}) = \sum_{(i,j,k)\in\Omega} [\mathcal{X}]_{i,j,k} \mathcal{E}_{i,j,k},$$

where $\mathcal{E}_{i,j,k}$ is a tensor in $\{0,1\}^{n_1 \times n_2 \times n_3}$ and all elements are zero except for the one at the position indexed by (i, j, k). The primary goal of the tensor completion problem is to reconstruct the tensor $\mathcal{T}$ from the entries on Ω. We utilize the approach proposed in the references [84, 138], which addresses the tensor completion issue through a specific convex optimization problem formulated as follows:

$$\begin{aligned} &\min_{\mathcal{X}} \|\mathcal{X}\|_{\mathrm{TNN}} \\ &\text{subject to } \mathcal{P}_\Omega(\mathcal{X}) = \mathcal{P}_\Omega(\mathcal{T}). \end{aligned} \tag{2.18}$$

Notice that TNN is convex but not strictly convex. Thus, there might be more than one local minimizer to the optimization problem (2.18). Therefore, we need to establish conditions to ensure that our optimization problem has a unique minimizer, which is exactly the tensor we seek to recover. The question of under what conditions $\mathcal{T}$ is the unique minimizer of the optimization problem (2.18) naturally arises. In response, Proposition 1 gives an affirmative answer. Before proceeding, it is important to highlight that in the following context, for convenience, we will interchangeably make use of $\|\cdot\|$ to denote the tensor spectral norm, tensor operator norm or the matrix spectral norm, depending on the specific situation.

Proposition 1 ([84])**.** *Assume that* $\mathcal{T} \in \mathbb{K}^{n_1 \times n_2 \times n_3}$ *of tubal-rank r satisfies the incoherence condition with parameter* μ_0 *and its compact t-SVD is given by* $\mathcal{T} = \mathcal{U} * \mathcal{S} * \mathcal{V}^\top$ *where* $\mathcal{U} \in \mathbb{K}^{n_1 \times r \times n_3}, \mathcal{S} \in \mathbb{K}^{r \times r \times n_3}$ *and* $\mathcal{V} \in \mathbb{K}^{n_2 \times r \times n_3}$*. Suppose that* Ω *is generated according to the Bernoulli sampling model*

with probability p. Then tensor $\mathcal{T}$ is a unique minimizer of the optimization problem (2.18)*, if the following two conditions hold:*

Condition 1. $\|\mathcal{P}_{\mathbb{T}}\mathcal{R}_{\Omega}\mathcal{P}_{\mathbb{T}} - \mathcal{P}_{\mathbb{T}}\| \leq \frac{1}{2}$

Condition 2. *There exists a tensor $\mathcal{Y}$ such that $\mathcal{P}_{\Omega}(\mathcal{Y}) = \mathcal{Y}$ and*

(a) $\|\mathcal{P}_{\mathbb{T}}(\mathcal{Y}) - \mathcal{U} * \mathcal{V}^{\top}\|_{\mathrm{F}} \leq \frac{1}{4}\sqrt{\frac{p}{n_3}}$

(b) $\|\mathcal{P}_{\mathbb{T}^{\perp}}(\mathcal{Y})\| \leq \frac{1}{2}$

Based on Proposition 1, our main result is derived through probabilistic estimation of the Condition 1 and Condition 2. Throughout this computation, we explicitly determine both the lower bound of the sampling probability p and the probability of the exact recovery of $\mathcal{T}$. The architecture of the entire proof is described as follows.

2.6.2.1 Architecture of the proof of Theorem 2.5

The proof of Theorem 2.5 follows the pipeline developed in [84, 138]. We first state a sufficient condition for $\mathcal{T}$ to be the unique optimal solution to the optimization problem (2.18) via constructing a dual certificate $\mathcal{Y}$ obeying two conditions. This result is summarized in Proposition 1. To obtain our main result Theorem 2.5, we just need to show that the conditions in Proposition 1 hold with a high probability. The Theorem 2.5 is built on the basis of Lemma 2.8, Lemma 2.9, Lemma 2.10 and Corollary 2.1. A detailed roadmap of the proof towards Theorem 2.5 is outlined in Figure 2.14.

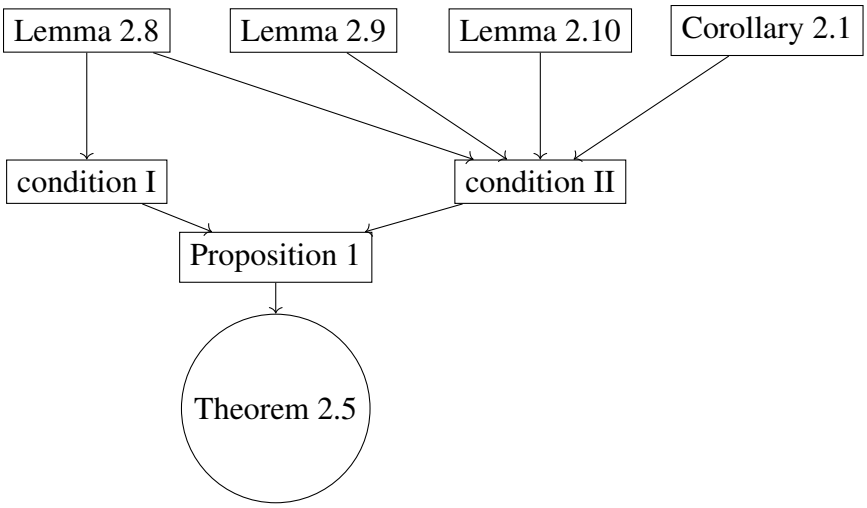


Figure 2.14 The structure of the proof of Theorem 2.5: The core of the proof for Theorem 2.5 relies on assessing the probability that certain conditions, specified in Proposition 1, are met. Condition I and Condition II serve as sufficient criteria to ensure the applicability of Proposition 1. Thus, the proof of Theorem 2.5 primarily involves determining the likelihood that condition I and II are satisfied. The probabilistic assessment of condition I utilizes Lemma 2.8 as a fundamental instrument. Similarly, the evaluation of condition II employs Lemmas 2.8 to 2.10, and Corollary 2.1 as essential tools.

Before delving into the proof of Theorem 2.5, we will introduce several supporting lemmas to lay the necessary foundation.

2.6.2.2 Supporting lemmas for the proof of Theorem 2.5

Lemma 2.5 (Non-commutative Bernstein inequality [115])**.** *Let $X_1, X_2, \cdots, X_L$ be independent zero-mean random matrices of dimension $n_1 \times n_2$. Suppose*

$$\sigma^2 = \max\left\{\left\|\mathbb{E}[\sum_{k=1}^{L} X_k X_k^\top]\right\|, \left\|\mathbb{E}[\sum_{k=1}^{L} X_k^\top X_k]\right\|\right\}$$

and $\|X_k\| \leq M$. Then for any $\tau \geq 0$,

$$\mathbb{P}\left(\left\|\sum_{k=1}^{L} X_k\right\| \geq \tau\right) \leq (n_1 + n_2)\exp\left(\frac{-\tau^2/2}{\sigma^2 + \frac{M\tau}{3}}\right).$$

The following lemma is a variant of Non-commutative Bernstein inequality.

Lemma 2.6. *Let $X_1, X_2, \cdots, X_L$ be independent zero-mean random matrices of dimension $n_1 \times n_2$. Suppose*

$$\sigma^2 = \max\left\{\left\|\mathbb{E}[\sum_{k=1}^{L} X_k X_k^\top]\right\|, \left\|\mathbb{E}[\sum_{k=1}^{L} X_k^\top X_k]\right\|\right\}$$

and $\|X_k\| \leq M$*, where* M *is a positive number. If we choose* $\tau = \sqrt{2c\sigma^2 \log(n_1+n_2)} + cM\log(n_1+n_2)$*, we have*

$$\mathbb{P}\left(\left\|\sum_{k=1}^{L} X_k\right\| \geq \sqrt{2c\sigma^2 \log(n_1+n_2)} + cM\log(n_1+n_2)\right) \leq (n_1+n_2)^{1-c},$$

where c *is any positive number greater than* 1*. The following fact is very useful and we will make frequent use of the result for the proofs of Theorem 2.5 and Proposition 1.*

Lemma 2.7. *Suppose* $\mathcal{T}$ *is an* $n_1 \times n_2 \times n_3$ *tensor with its compact t-SVD given by* $\mathcal{T} = \mathcal{U} * \Sigma * \mathcal{V}^\top$ *and satisfy incoherence condition with parameter* μ_0*. Then,*

$$\|\mathcal{P}_{\mathbb{T}}(\mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^\top)\|_{\mathrm{F}}^2 \leq \frac{(n_1+n_2)\mu_0 r}{n_1 n_2}.$$

The following lemma shows how likely that the operator norm $\mathcal{P}_{\mathbb{T}}\mathcal{R}_\Omega\mathcal{P}_{\mathbb{T}} - \mathcal{P}_{\mathbb{T}}$ is smaller than $\frac{1}{2}$. Such result will help us calculate how likely the Condition 1 in Proposition 1 holds.

Lemma 2.8. *Assume that* Ω *is generated according to the Bernoulli distribution with probability* p*, then*

$$\|\mathcal{P}_{\mathbb{T}}\mathcal{R}_\Omega\mathcal{P}_{\mathbb{T}} - \mathcal{P}_{\mathbb{T}}\| \leq \frac{1}{2}$$

holds with probability at least $1 - 2n_1n_2n_3 \exp\left(-\frac{3pn_1n_2}{28(n_1+n_2)\mu_0 r}\right)$.

The following lemma states that given an arbitrary tensor $\mathcal{X} \in \mathbb{K}^{n_1\times n_2\times n_3}$, tensor spectral norm of difference between $\mathcal{R}_\Omega(\mathcal{X})$ and $\mathcal{X}$ can be bounded with tensor infinity norm and $l_{\infty,2}$ norm with a high probability.

Lemma 2.9. *Given an arbitrary tensor* $\mathcal{X} \in \mathbb{K}^{n_1\times n_2\times n_3}$*. Assume that* Ω *is generated according to the Bernoulli distribution with probability* p*. Then, for any constant* $c_2 > 1$*, we have*

$$\|\mathcal{R}_\Omega(\mathcal{X}) - \mathcal{X}\| \leq \|\mathcal{X}\|_{\infty,2}\sqrt{\frac{2c_2}{p}\log((n_1+n_2)n_3)} + \frac{c_2\log((n_1+n_2)n_3)}{p}\|\mathcal{X}\|_\infty \tag{2.19}$$

holds with probability at least $1 - ((n_1+n_2)n_3)^{1-c_2}$.

The following lemma states that given an arbitrary tensor $\mathcal{X} \in \mathbb{K}^{n_1\times n_2\times n_3}$, the bound of $l_{\infty,2}$ distance between $\mathcal{P}_{\mathbb{T}}\mathcal{R}_\Omega(\mathcal{X})$ and $\mathcal{P}_{\mathbb{T}}(\mathcal{X})$ can be controlled by the $l_{\infty,2}$ norm of $\mathcal{X}$ and the tensor infinity norm of $\mathcal{X}$ with a high probability.

Lemma 2.10. *Assume that Ω is generated according to the Bernoulli distribution with probability p. For any positive number $c_1 \geq 2$, then we can get*

$$\mathbb{P}\left(\|(\mathcal{P}_{\mathbb{T}}\mathcal{R}_{\Omega}(\mathcal{X}) - \mathcal{P}_{\mathbb{T}}(\mathcal{X}))\|_{\infty,2} \leq \sqrt{\frac{4c_1(n_1+n_2)\mu_0 r \log\left((n_1+n_2)n_3\right)}{pn_1n_2}} \cdot \|\mathcal{X}\|_{\infty,2} \right.$$
$$\left. + \frac{c_1 \log((n_1+n_2)n_3)}{p}\sqrt{\frac{(n_1+n_2)\mu_0 r}{n_1n_2}}\|\mathcal{X}\|_{\infty} \right) \geq 1 - ((n_1+n_2)n_3)^{2-c_1}.$$

The following lemma states that, given an arbitrary tensor $\mathcal{X} \in \mathbb{R}^{n_1\times n_2\times n_3}$, the tensor infinity norm of $\mathcal{P}_{\mathbb{T}}\mathcal{R}_{\Omega}\mathcal{P}_{\mathbb{T}}(\mathcal{X}) - \mathcal{P}_{\mathbb{T}}(\mathcal{X})$ can be bounded by the tensor infinity norm of $\mathcal{P}_{\mathbb{T}}(\mathcal{X})$ with a high probability.

Lemma 2.11. *Assume that Ω is generated according to the Bernoulli distribution with probability p. For any $\mathcal{X} \in \mathbb{K}^{n_1\times n_2\times n_3}$, then*

$$\|\left(\mathcal{P}_{\mathbb{T}}\mathcal{R}_{\Omega}\mathcal{P}_{\mathbb{T}} - \mathcal{P}_{\mathbb{T}}\right)(\mathcal{X})\|_{\infty} \leq \frac{1}{2}\|\mathcal{P}_{\mathbb{T}}(\mathcal{X})\|_{\infty}$$

holds with probability at least $1 - 2n_1n_2n_3 \exp\left(\frac{-3pn_1n_2}{16(n_1+n_2)\mu_0 r}\right)$.

When $\mathcal{P}_{\mathbb{T}}(\mathcal{X}) = \mathcal{X}$, we can easily achieve the following corollary.

Corollary 2.1. *Assume that Ω is generated according to the Bernoulli distribution with probability q. For any $\mathcal{X} \in \mathbb{K}^{n_1\times n_2\times n_3}$, if $\mathcal{P}_{\mathbb{T}}(\mathcal{X}) = \mathcal{X}$ then*

$$\left\|\left(\mathcal{P}_{\mathbb{T}}\mathcal{R}_{\Omega_t}\mathcal{P}_{\mathbb{T}} - \mathcal{P}_{\mathbb{T}}\right)(\mathcal{X})\right\|_{\infty} \leq \frac{1}{2}\|\mathcal{X}\|_{\infty}$$

holds with probability at least $1 - 2n_1n_2n_3 \exp\left(\frac{-3qn_1n_2}{28(n_1+n_2)\mu_0 r}\right)$.

Corollary 2.1 is used to give a probabilistic estimate towards the lower bound of $\|\mathcal{D}_t\|_{\infty}$, where $\mathcal{D}_t$ is defined in Equation (2.22) later. Now we are ready to provide the proof of Theorem 2.5.

Proof of Theorem 2.5. First of all, one can get that the Condition 1 holds with probability at least

$$1 - 2n_1n_2n_3 \exp\left(-\frac{3pn_1n_2}{28(n_1+n_2)\mu_0 r}\right) \tag{2.20}$$

according to Lemma 2.8.

Next, our main goal is to construct a dual certificate $\mathcal{Y}$ that satisfies the condition 2. We do this using the Golfing Scheme [23, 50]. Choose t_0 as

$$t_0 \geq \left\lceil \log_2\left(4\sqrt{\frac{n_3 r}{p}}\right)\right\rceil \tag{2.21}$$

where $\lceil\cdot\rceil$ is the ceil function. Suppose that the set Ω of observed entries is generated from $\Omega = \cup_{t=1}^{t_0}\Omega_t$ with $\mathbb{P}[(i,j,k) \in \Omega_t] = q := 1-(1-p)^{\frac{1}{t_0}}$ and is independent of each other. It is easy to see that for any $(i,j,k) \in [n_1]\times[n_2]\times[n_3]$,

$$\begin{aligned}\mathbb{P}[(i,j,k)\in\Omega] =& 1-\mathbb{P}[(i,j,k)\notin\cup_{t=1}^{t_0}\Omega_t]\\ =& 1-\prod_{t=1}^{t_0}\mathbb{P}[(i,j,k)\in\Omega_t^c] = 1-\prod_{t=0}^{t_0}(1-p)^{\frac{1}{t_0}} = p.\end{aligned}$$

Therefore, the construction of $\Omega = \bigcup_{t=1}^{t_0}\Omega_t$ shares the same distribution as that of Ω. Let $\{\mathcal{A}_t \in \mathbb{K}^{n_1\times n_2\times n_3} : t = 0,\cdots,t_0\}$ be a sequence of tensors with $\mathcal{A}_0 = \mathbf{0}$ and

$$\mathcal{A}_t = \mathcal{A}_{t-1} + \mathcal{R}_{\Omega_t}\mathcal{P}_{\mathbb{T}}(\mathcal{U}*\mathcal{V}^\top - \mathcal{P}_{\mathbb{T}}(\mathcal{A}_{t-1})),$$

where $\mathcal{R}_{\Omega_t}(\mathcal{T}) := \sum_{i\in[n_1],j\in[n_2],k\in[n_3]}\frac{1}{q}1_{\Omega_t}(i,j,k)[\mathcal{T}]_{i,j,k}\mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^\top$. Set $\mathcal{Y} := \mathcal{A}_{t_0}$.

Next, our goal is to prove that $\mathcal{P}_\Omega(\mathcal{Y}) = \mathcal{Y}$ by mathematical induction. For $t = 0$, $\mathcal{P}_\Omega(\mathcal{A}_0) = \mathcal{P}_\Omega(\mathbf{0}) = \mathbf{0} = \mathcal{A}_0$. Notice that

$$\begin{aligned}\mathcal{A}_1 &= \mathcal{A}_0 + \mathcal{R}_{\Omega_1}\mathcal{P}_{\mathbb{T}}(\mathcal{U}*\mathcal{V}^\top - \mathcal{P}_{\mathbb{T}}(\mathcal{A}_0))\\ &= \mathcal{A}_0 + \mathcal{R}_{\Omega_1}\mathcal{P}_{\mathbb{T}}(\mathcal{U}*\mathcal{V}^\top)\\ &= \mathcal{R}_{\Omega_1}(\mathcal{U}*\mathcal{U}^\top*(\mathcal{U}*\mathcal{V}^\top) + (\mathcal{U}*\mathcal{V}^\top)*\mathcal{V}*\mathcal{V}^\top - \mathcal{U}*\mathcal{U}^\top*(\mathcal{U}*\mathcal{V}^\top)*\mathcal{V}*\mathcal{V}^\top)\\ &= \mathcal{R}_{\Omega_1}(\mathcal{U}*\mathcal{V}^\top).\end{aligned}$$

Due to $\Omega_1 \subseteq \Omega$, it is easy to see that $\mathcal{P}_\Omega(\mathcal{A}_1) = \mathcal{P}_\Omega(\mathcal{R}_{\Omega_1}(\mathcal{U}*\mathcal{V}^\top)) = \mathcal{R}_{\Omega_1}(\mathcal{U}*\mathcal{V}^\top) = \mathcal{A}_1$. Assume that for $k \le t_0 - 1$, it holds that $\mathcal{P}_\Omega(\mathcal{A}_k) = \mathcal{A}_k$. By linearity of operator $\mathcal{P}_\Omega$ and $\Omega_{t_0} \subseteq \Omega$, it follows that

$$\begin{aligned}\mathcal{P}_\Omega(\mathcal{Y}) &= \mathcal{P}_\Omega(\mathcal{A}_{t_0})\\ &= \mathcal{P}_\Omega(\mathcal{A}_{t_0-1} + \mathcal{R}_{\Omega_{t_0}}\mathcal{P}_{\mathbb{T}}(\mathcal{U}*\mathcal{V}^\top - \mathcal{P}_{\mathbb{T}}(\mathcal{A}_{t_0-1})))\\ &= \mathcal{P}_\Omega(\mathcal{A}_{t_0-1}) + \mathcal{P}_\Omega(\mathcal{R}_{\Omega_{t_0}}\mathcal{P}_{\mathbb{T}}(\mathcal{U}*\mathcal{V}^\top - \mathcal{P}_{\mathbb{T}}(\mathcal{A}_{t_0-1})))\\ &= \mathcal{A}_{t_0-1} + \mathcal{R}_{\Omega_{t_0}}\mathcal{P}_{\mathbb{T}}(\mathcal{U}*\mathcal{V}^\top - \mathcal{P}_{\mathbb{T}}(\mathcal{A}_{t_0-1})) = \mathcal{A}_{t_0} = \mathcal{Y}.\end{aligned}$$

Therefore $\mathcal{Y} = \mathcal{A}_{t_0}$ is the dual certificate.

Now let's prove that $\|\mathcal{P}_{\mathbb{T}}(\mathcal{Y}) - \mathcal{U} * \mathcal{V}^\top\|_{\mathrm{F}} \le \frac{1}{4}\sqrt{\frac{p}{n_3}}$. For $t = 0, 1, \cdots, t_0$, set

$$\mathcal{D}_t = \mathcal{U} * \mathcal{V}^\top - \mathcal{P}_{\mathbb{T}}(\mathcal{A}_t). \tag{2.22}$$

Notice that

$$q = 1 - (1-p)^{\frac{1}{t_0}} \ge 1 - (1 - \frac{p}{t_0}) = \frac{p}{t_0}. \tag{2.23}$$

Thus, one can derive the following results by Lemma 2.8: for each t,

$$\|\mathcal{D}_t\|_{\mathrm{F}} \le \left\|\mathcal{P}_{\mathbb{T}} - \mathcal{P}_{\mathbb{T}}\mathcal{R}_{\Omega_t}\mathcal{P}_{\mathbb{T}}\right\| \|\mathcal{D}_{t-1}\|_{\mathrm{F}} \le \frac{1}{2}\|\mathcal{D}_{t-1}\|_{\mathrm{F}} \tag{2.24}$$

holds with probability at least

$$1 - 2n_1n_2n_3 \exp(-\frac{3qn_1n_2}{28(n_1+n_2)\mu_0 r}).$$

Applying (2.24) from $t = t_0$ to $t = 1$, we will have that

$$\|\mathcal{P}_{\mathbb{T}}(\mathcal{Y} - \mathcal{U} * \mathcal{V}^\top)\|_{\mathrm{F}} = \|\mathcal{D}_{t_0}\|_{\mathrm{F}} \le \frac{1}{2}\|\mathcal{D}_{t_0-1}\|_{\mathrm{F}} \le \cdots \le (\frac{1}{2})^{t_0}\|\mathcal{U} * \mathcal{V}^\top\|_{\mathrm{F}} \le (\frac{1}{2})^{t_0}\sqrt{r} \tag{2.25}$$

holds with probability at least

$$1 - 2t_0n_1n_2n_3 \exp(-\frac{3qn_1n_2}{28(n_1+n_2)\mu_0 r}).$$

Since $t_0 \ge \left\lceil \log_2\left(4\sqrt{\frac{n_3 r}{p}}\right)\right\rceil$, $\|\mathcal{P}_{\mathbb{T}}(\mathcal{Y} - \mathcal{U} * \mathcal{V}^\top)\|_{\mathrm{F}} \le \frac{1}{4}\sqrt{\frac{p}{n_3}}$ holds with probability at least

$$1 - 2t_0n_1n_2n_3 \exp(-\frac{3qn_1n_2}{28(n_1+n_2)\mu_0 r}). \tag{2.26}$$

Next, we move on to prove $\|\mathcal{P}_{\mathbb{T}^\perp}(\mathcal{Y})\| \le \frac{1}{2}$. Recall that $\mathcal{Y} = \sum_{i=1}^{t_0} \mathcal{R}_{\Omega_t}\mathcal{P}_{\mathbb{T}}\mathcal{D}_{t-1}$. By applying Lemma 2.9 for t_0 times, we can get

$$\begin{aligned}
&\|\mathcal{P}_{\mathbb{T}^\perp}(\mathcal{Y})\| \\
&\le \sum_{t=1}^{t_0} \left\|\mathcal{P}_{\mathbb{T}^\perp}\left(\mathcal{R}_{\Omega_t}\mathcal{P}_{\mathbb{T}} - \mathcal{P}_{\mathbb{T}}\right)(\mathcal{D}_{t-1})\right\| \\
&\le \sum_{t=1}^{t_0} \left\|\left(\mathcal{R}_{\Omega_t} - \mathcal{I}\right)(\mathcal{P}_{\mathbb{T}}(\mathcal{D}_{t-1}))\right\| \\
&\le \sum_{t=1}^{t_0} \left(\frac{c_2\log((n_1+n_2)n_3)}{q}\|\mathcal{P}_{\mathbb{T}}(\mathcal{D}_{t-1})\|_\infty + \sqrt{\frac{2c_2\log((n_1+n_2)n_3)}{q}}\|\mathcal{P}_{\mathbb{T}}(\mathcal{D}_{t-1})\|_{\infty,2}\right) \qquad (2.27) \\
&\le \sum_{t=1}^{t_0} \left(\frac{c_2\log((n_1+n_2)n_3)}{q}\|\mathcal{P}_{\mathbb{T}}(\mathcal{D}_{t-1})\|_\infty + \sqrt{\frac{2c_2\log((n_1+n_2)n_3)}{q}}\|\mathcal{P}_{\mathbb{T}}(\mathcal{D}_{t-1})\|_{\infty,2}\right) \qquad (2.28) \\
&= \sum_{t=1}^{t_0} \left(\frac{c_2\log((n_1+n_2)n_3)}{q}\|\mathcal{D}_{t-1}\|_\infty + \sqrt{\frac{2c_2\log((n_1+n_2)n_3)}{q}}\|\mathcal{D}_{t-1}\|_{\infty,2}\right) \qquad (2.29)
\end{aligned}$$

holds with probability at least

$$1 - \frac{t_0}{((n_1+n_2)n_3)^{c_2-1}}. \tag{2.30}$$

(2.28) holds due to (2.23). (2.29) is due to $\mathcal{P}_{\mathbb{T}}(\mathcal{D}_t) = \mathcal{D}_t$ by the construction of $\mathcal{D}_t$ in Equation (2.22).

Next, we will bound (2.29) by bounding these two terms:

(i) $\sum_{t=1}^{t_0} \frac{c_2 \log((n_1+n_2)n_3)}{q} \|\mathcal{D}_{t-1}\|_\infty$

(ii) $\sum_{t=1}^{t_0} \sqrt{\frac{2c_2 \log((n_1+n_2)n_3)}{q}} \|\mathcal{D}_{t-1}\|_{\infty,2}$

via estimating the upper bounds of $\|\mathcal{D}_{t-1}\|_\infty$ and $\|\mathcal{D}_{t-1}\|_{\infty,2}$.

By applying Corollary 2.1 for $t-1$ times, where $2 \le t \le t_0$, we have

$$\begin{aligned}\|\mathcal{D}_{t-1}\|_\infty &= \left\|\left(\mathcal{P}_{\mathbb{T}} - \mathcal{P}_{\mathbb{T}}\mathcal{R}_{\Omega_{t-1}}\mathcal{P}_{\mathbb{T}}\right)\cdots\left(\mathcal{P}_{\mathbb{T}} - \mathcal{P}_{\mathbb{T}}\mathcal{R}_{\Omega_1}\mathcal{P}_{\mathbb{T}}\right)\mathcal{D}_0\right\|_\infty \\ &\le \left(\frac{1}{2}\right)^{t-1}\|\mathcal{D}_0\|_\infty\end{aligned}$$

holds with probability at least $1 - 2n_1n_2n_3(t-1)\exp\left(\frac{-3qn_1n_2}{28(n_1+n_2)\mu_0 r}\right)$. Therefore,

$$\sum_{t=1}^{t_0} \frac{c_2 \log((n_1+n_2)n_3)}{q}\|\mathcal{D}_{t-1}\|_\infty \le \frac{2c_2 \log((n_1+n_2)n_3)}{q} \cdot \|\mathcal{D}_0\|_\infty$$

holds with probability at least

$$1 - 2n_1n_2n_3(t_0-1)\exp\left(\frac{-3qn_1n_2}{28(n_1+n_2)\mu_0 r}\right). \tag{2.31}$$

Now we are going to estimate the upper bound for $\sum_{t=1}^{t_0} \sqrt{\frac{2c_2 \log((n_1+n_2)n_3)}{q}} \|\mathcal{D}_{t-1}\|_{\infty,2}$ by bounding $\|\mathcal{D}_{t-1}\|_{\infty,2}$. For simplicity of expression, we will denote

$a = 2\cdot\sqrt{\frac{c_1 \log((n_1+n_2)n_3)}{q}}\sqrt{\frac{(n_1+n_2)\mu_0 r}{n_1n_2}}$ and $b = \frac{c_1(\log((n_1+n_2)n_3))}{q}\sqrt{\frac{(n_1+n_2)\mu_0 r}{n_1n_2}}$. By applying Lemma 2.10 by $t-1$ times and considering the fact that $\mathcal{P}_{\mathbb{T}}(\mathcal{D}_s) = \mathcal{D}_s$ for all $0 \le s \le t_0$, we obtain

$$\begin{aligned}\|\mathcal{D}_{t-1}\|_{\infty,2} =& \left\|\left(\mathcal{P}_{\mathbb{T}} - \mathcal{P}_{\mathbb{T}}\mathcal{R}_{\Omega_{t-1}}\mathcal{P}_{\mathbb{T}}\right)(\mathcal{D}_{t-2})\right\|_{\infty,2} \\ \le& a\|\mathcal{D}_{t-2}\|_{\infty,2} + b\|\mathcal{D}_{t-2}\|_\infty \le \cdots \le a^{t-1}\|\mathcal{D}_0\|_{\infty,2} + b\sum_{i=0}^{t-2} a^i\|\mathcal{D}_{t-2-i}\|_\infty\end{aligned}$$

holds with probability at least $1 - \frac{t-1}{((n_1n_3+n_2n_3)^{c_1-2}}$ for $2 \le t \le t_0$.

Therefore,

$$\begin{aligned}
&\sum_{t=1}^{t_0}\sqrt{\frac{2c_2\log((n_1+n_2)n_3)}{q}}\,\|\mathcal{D}_{t-1}\|_{\infty,2}\\
\leq&\sqrt{\frac{2c_2\log((n_1+n_2)n_3)}{q}}\left(\left(\sum_{t=1}^{t_0}a^{t-1}\|\mathcal{D}_0\|_{\infty,2}\right)+\sum_{t=2}^{t_0}b\sum_{i=0}^{t-2}a^i\|\mathcal{D}_{t-2-i}\|_{\infty}\right)\\
=&\sqrt{\frac{2c_2\log((n_1+n_2)n_3)}{q}}\cdot\left(\|\mathcal{D}_0\|_{\infty,2}\frac{1-a^{t_0}}{1-a}+b\cdot\sum_{t=2}^{t_0}\sum_{i=0}^{t-2}a^i\|\mathcal{D}_{t-2-i}\|_{\infty}\right)
\end{aligned}$$

holds with probability at least

$$1-\frac{t_0-1}{(n_1n_3+n_2n_3)^{c_1-2}}.$$

Taking the process of estimating the upper bound for $\|\mathcal{D}_t\|_\infty$ into account, we thus have

$$\begin{aligned}
(2.29)\leq&\frac{2c_2\log((n_1+n_2)n_3)}{q}\cdot\|\mathcal{D}_0\|_\infty+\sqrt{\frac{2c_2\log((n_1+n_2)n_3)}{q}}\cdot\frac{\|\mathcal{D}_0\|_{\infty,2}}{1-a}\\
&+\sqrt{\frac{2c_2\log((n_1+n_2)n_3)}{q}}\cdot b\cdot\sum_{t=2}^{t_0}\sum_{i=0}^{t-2}a^i\left(\frac{1}{2}\right)^{t-2-i}\|\mathcal{D}_0\|_\infty \qquad (2.32)\\
\leq&\frac{2c_2\log((n_1+n_2)n_3)}{q}\cdot\|\mathcal{D}_0\|_\infty+\sqrt{\frac{2c_2\log((n_1+n_2)n_3)}{q}}\cdot\frac{\|\mathcal{D}_0\|_{\infty,2}}{1-a} \qquad (2.33)\\
&+\sqrt{\frac{2c_2\log((n_1+n_2)n_3)}{q}}\cdot\frac{2b}{1-2a}\cdot\|\mathcal{D}_0\|_\infty \qquad (2.34)
\end{aligned}$$

holds with probability at least $1-\frac{t_0-1}{(n_1n_3+n_2n_3)^{c_1-2}}-2(t_0-1)n_1n_2n_3\exp\left(\frac{-3qn_1n_2}{28(n_1+n_2)\mu_0r}\right)$ when $0<a\leq\frac{1}{4}$.

Therefore,

$$\begin{aligned}
\|\mathcal{P}_{\mathbb{T}^\perp}(\mathcal{Y})\|\leq&\sum_{t=1}^{t_0}\left(\frac{c_2\log((n_1+n_2)n_3)}{q}\|\mathcal{D}_{t-1}\|_\infty+\sqrt{\frac{2c_2\log((n_1+n_2)n_3)}{q}}\,\|\mathcal{D}_{t-1}\|_{\infty,2}\right)\\
\leq&\frac{2c_2\log((n_1+n_2)n_3)}{q}\cdot\|\mathcal{D}_0\|_\infty+\sqrt{\frac{2c_2\log((n_1+n_2)n_3)}{q}}\cdot\frac{\|\mathcal{D}_0\|_{\infty,2}}{1-a}\\
&+\sqrt{\frac{2c_2\log((n_1+n_2)n_3)}{q}}\cdot\frac{2b}{1-2a}\cdot\|\mathcal{D}_0\|_\infty
\end{aligned}$$

holds with probability at least

$$1-\frac{t_0-1}{(n_1n_3+n_2n_3)^{c_1-2}}-2(t_0-1)n_1n_2n_3\exp\left(\frac{-3qn_1n_2}{28(n_1+n_2)\mu_0r}\right)-\frac{t_0}{(n_1n_3+n_2n_3)^{c_2-1}}$$

provided that $0<a\leq\frac{1}{4}$.

Note that $\|\mathcal{D}_0\|_\infty = \|\mathcal{U} * \mathcal{V}^\top\|_\infty \leq \frac{(n_1+n_2)\mu_0 r}{2n_1n_2}$ and $\|\mathcal{D}_0\|_{\infty,2} = \|\mathcal{U} * \mathcal{V}^\top\|_{\infty,2} \leq \sqrt{\frac{(n_1+n_2)\mu_0 r}{n_1n_2}}$.

And combining (2.30), (2.31), and (2.6.2.2), we thus have

$$\begin{aligned}
\|\mathcal{P}_{\mathbb{T}^\perp}(\mathcal{Y})\| \leq & \frac{c_2\log((n_1+n_2)n_3)}{q}\cdot\frac{(n_1+n_2)\mu_0 r}{n_1n_2} + \sqrt{\frac{2c_2\log((n_1+n_2)n_3)}{q}}\cdot\frac{\sqrt{(n_1+n_2)\mu_0 r}}{(1-a)\sqrt{n_1n_2}} \\
& + \sqrt{\frac{2c_2\log((n_1+n_2)n_3)}{q}}\cdot\frac{b}{1-2a}\cdot\frac{(n_1+n_2)\mu_0 r}{n_1n_2} \\
\leq & \frac{c_2\log((n_1+n_2)n_3)}{q}\cdot\frac{(n_1+n_2)\mu_0 r}{n_1n_2} + \frac{4\sqrt{2c_2}}{3}\cdot\sqrt{\frac{\log((n_1+n_2)n_3)}{q}\cdot\frac{(n_1+n_2)\mu_0 r}{n_1n_2}} \\
& + 2c_1\sqrt{2c_2}\left(\frac{\log((n_1+n_2)n_3)}{q}\cdot\frac{(n_1+n_2)\mu_0 r}{n_1n_2}\right)^{3/2}
\end{aligned}$$

holds with probability at least

$$1 - \frac{t_0-1}{(n_1n_3+n_2n_3)^{c_1-2}} - 2(t_0-1)n_1n_2n_3\exp\left(\frac{-3qn_1n_2}{28(n_1+n_2)\mu_0 r}\right) - \frac{t_0}{(n_1n_3+n_2n_3)^{c_2-1}}$$

provided that $0 < a \leq \frac{1}{4}$.

Since $a = 2\cdot\sqrt{\frac{c_1\log((n_1+n_2)n_3)}{q}}\sqrt{\frac{(n_1+n_2)\mu_0 r}{n_1n_2}}$, the restriction $0 < a \leq \frac{1}{4}$ is equivalent to

$$q \geq 64c_1\log((n_1+n_2)n_3)\cdot\frac{(n_1+n_2)\mu_0 r}{n_1n_2}.$$

Notice that

$$p \geq \frac{256(n_1+n_2)\mu_0\beta r\log^2((n_1+n_2)n_3)}{n_1n_2}.$$

We thus have

$$\begin{aligned}
t_0 = & \lceil\log_2\left(4\sqrt{\frac{n_3 r}{p}}\right)\rceil \\
\leq & \left\lceil\log_2\left(4\sqrt{\frac{n_1n_2n_3}{256(n_1+n_2)\mu_0\beta\log^2((n_1+n_2)n_3)}}\right)\right\rceil \\
\leq & \left\lceil\frac{1}{2}\log_2\left(\frac{n_1n_2n_3}{(n_1+n_2)\mu_0\beta}\right) - 2\right\rceil \leq \log((n_1+n_2)n_3)
\end{aligned}$$

In addition $q \geq \frac{p}{t_0}$, we have

$$\begin{aligned}
q \geq & \frac{256(n_1+n_2)\mu_0\beta r\log^2((n_1+n_2)n_3)}{n_1n_2}\cdot\frac{1}{\log((n_1+n_2)n_3)} \\
= & \frac{256(n_1+n_2)\mu_0\beta r\log((n_1+n_2)n_3)}{n_1n_2},
\end{aligned}$$

i.e., $\frac{(n_1+n_2)\mu_0 r\log((n_1+n_2)n_3)}{qn_1n_2} \leq \frac{1}{256\beta}$.

Therefore the condition that $q \geq 64c_1\log((n_1+n_2)n_3)\frac{(n_1+n_2)\mu_0 r}{n_1n_2}$ holds when $c_1 = 4\beta$. Hence, the

condition $a \leq \frac{1}{4}$ holds. In addition, by setting $c_2 = 12\beta$, we have

$$\begin{aligned}\|\mathcal{P}_{\mathbb{T}^\perp}(\mathcal{Y})\| \leq & \frac{c_2 \log((n_1+n_2)n_3)}{q} \cdot \frac{(n_1+n_2)\mu_0 r}{n_1 n_2} + \frac{4\sqrt{2c_2}}{3} \cdot \sqrt{\frac{\log((n_1+n_2)n_3)}{q} \cdot \frac{(n_1+n_2)\mu_0 r}{n_1 n_2}} \\ & + 2c_1\sqrt{2c_2}\left(\frac{\log((n_1+n_2)n_3)}{q} \cdot \frac{(n_1+n_2)\mu_0 r}{n_1 n_2}\right)^{3/2} \\ \leq & \frac{c_2}{256\beta} + \frac{4\sqrt{2c_2}}{3}\sqrt{\frac{1}{256\beta}} + 2c_1\sqrt{2c_2}\left(\frac{1}{256\beta}\right)^{3/2} < \frac{1}{2}.\end{aligned}$$

with probability at least

$$\begin{aligned}& 1 - \frac{\log(n_1 n_3 + n_2 n_3)}{(n_1 n_3 + n_2 n_3)^{4\beta-2}} - \frac{\log(n_1 n_3 + n_2 n_3)}{(n_1 n_3 + n_2 n_3)^{27\beta-2}} - \frac{\log(n_1 n_3 + n_2 n_3)}{(n_1 n_3 + n_2 n_3)^{12\beta-1}} \\ & \geq 1 - \frac{3\log(n_1 n_3 + n_2 n_3)}{(n_1 n_3 + n_2 n_3)^{4\beta-2}}.\end{aligned}$$

Notice that the probabilistic estimation for the validity of the Condition 2 is predicated on the assumption that the Condition 1 holds true, where we show $\|\mathcal{D}\|_\infty \leq (\frac{1}{2})^{t-1}\|\mathcal{D}_0\|_\infty$ based on the Condition 1. Thus, $\mathcal{T}$ is the unique minimizer with probability at least

$$1 - \frac{3\log(n_1 n_3 + n_2 n_3)}{(n_1 n_3 + n_2 n_3)^{4\beta-2}}.$$

□

2.6.2.3 Proof of supporting lemmas towards Theorem 2.5

Proof of Lemma 2.6. Substitute $\tau = \sqrt{2c\sigma^2 \log(n_1+n_2)} + cM\log(n_1+n_2)$ to $\frac{-\tau^2/2}{\sigma^2 + \frac{M\tau}{3}}$ in Lemma 2.5. We can get

$$\begin{aligned}\frac{-\tau^2/2}{\sigma^2 + \frac{M\tau}{3}} &= -\frac{2c\sigma^2\log(n_1+n_2) + 2\sqrt{2}c^{\frac{3}{2}}\sigma M\log^{\frac{3}{2}}(n_1+n_2) + c^2M^2\log^2(n_1+n_2)}{2\sigma^2 + \frac{2\sqrt{2}}{3}c^{\frac{1}{2}}\sigma M\log^{\frac{1}{2}}(n_1+n_2) + \frac{2cM^2}{3}\log(n_1+n_2)} \\ &\leq -c\log(n_1+n_2).\end{aligned}$$

□

Proof of Lemma 2.7.

$$\begin{aligned}\|\mathcal{P}_{\mathbb{T}}(\mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^\top)\|_{\mathrm{F}}^2 &= \langle \mathcal{P}_{\mathbb{T}}(\mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^\top), \mathcal{P}_{\mathbb{T}}(\mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^\top)\rangle \\ &= \langle \mathcal{P}_{\mathbb{T}}(\mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^\top), \mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^\top\rangle \\ &= \left\|\mathcal{U}^\top * \mathring{\mathbf{e}}_i\right\|_{\mathrm{F}}^2 + \left\|\mathcal{V}^\top * \mathring{\mathbf{e}}_j\right\|_{\mathrm{F}}^2 - \left\|\mathcal{U}^\top * \mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^\top * \mathcal{V}\right\|_{\mathrm{F}}^2, \\ &\leq \left\|\mathcal{U}^\top * \mathring{\mathbf{e}}_i\right\|_{\mathrm{F}}^2 + \left\|\mathcal{V}^\top * \mathring{\mathbf{e}}_j\right\|_{\mathrm{F}}^2 = \frac{(n_1+n_2)\mu_0 r}{n_1 n_2}\end{aligned}$$

□

In the following context, we use $\delta_{i,j,k}$ to represent indicator function $\mathbf{1}_{(i,j,k)\in\Omega}$.

Proof of Lemma 2.8. By the fact that $\mathcal{P}_{\mathbb{T}}$ is self adjoint and idempotent operator, we can get $\mathbb{E}[\mathcal{P}_{\mathbb{T}}\mathcal{R}_{\Omega}\mathcal{P}_{\mathbb{T}}] = \mathcal{P}_{\mathbb{T}}(\mathbb{E}\mathcal{R}_{\Omega})\mathcal{P}_{\mathbb{T}} = \mathcal{P}_{\mathbb{T}}$. It is easy to check that

$$\mathcal{P}_{\mathbb{T}}\mathcal{R}_{\Omega}\mathcal{P}_{\mathbb{T}}(\mathcal{X}) = \sum_{i,j,k}\frac{1}{p}\delta_{i,j,k}\left\langle \mathcal{X}, \mathcal{P}_{\mathbb{T}}\left(\mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^{\top}\right)\right\rangle \mathcal{P}_{\mathbb{T}}\left(\mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^{\top}\right).$$

Fix a tensor $\mathcal{X} \in \mathbb{K}^{n_1\times n_2\times n_3}$, we can write

$$\begin{aligned}\left(\mathcal{P}_{\mathbb{T}}\mathcal{R}_{\Omega}\mathcal{P}_{\mathbb{T}} - \mathcal{P}_{\mathbb{T}}\right)(\mathcal{X}) &= \sum_{i,j,k}\left(\frac{1}{p}\delta_{ijk} - 1\right)\left\langle \mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^{\top}, \mathcal{P}_{\mathbb{T}}(\mathcal{X})\right\rangle \mathcal{P}_{\mathbb{T}}\left(\mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^{\top}\right)\\ &=: \sum_{i,j,k}\mathcal{H}_{ijk}(\mathcal{X})\end{aligned}$$

where $\mathcal{H}_{ijk} : \mathbb{K}^{n_1\times n_2\times n_3} \to \mathbb{K}^{n_1\times n_2\times n_3}$ is a self-adjoint random operator and $\delta_{i,j,k}$ is the indicator function. It is direct to see that $\mathbb{E}\left[\mathcal{H}_{ijk}\right] = 0$ due to the fact that $\mathbb{E}(\frac{1}{p}\delta_{i,j,k} - 1) = \frac{1}{p}\mathbb{E}(\delta_{i,j,k}) - 1 = 0$. Define the operator $\overline{\mathcal{H}}_{ijk} : \mathbb{B} \to \mathbb{B}$, where $\mathbb{B} = \left\{\overline{\mathcal{B}} : \mathcal{B} \in \mathbb{K}^{n_1\times n_2\times n_3}\right\}$ denotes the set consists of block diagonal matrices with the blocks as the frontal slices of $\overline{\mathcal{B}}$, as

$$\overline{\mathcal{H}}_{ijk}(\overline{\mathcal{X}}) := \overline{\mathcal{H}_{ijk}(\mathcal{X})} = (\frac{1}{p}\delta_{ijk} - 1)\left\langle \mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^{\top}, \mathcal{P}_{\mathbb{T}}(\mathcal{X})\right\rangle \overline{\mathcal{P}_{\mathbb{T}}\left(\mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^{\top}\right)}.$$

It is easy to check that $\overline{\mathcal{H}}_{ijk}$ is also self-adjoint by using the fact that the operator $\mathcal{P}_{\mathbb{T}}(\cdot)$ is self-adjoint. Using the fact that $\mathbb{E}(\frac{1}{p}\delta_{i,j,k} - 1) = 0$ again, we have $\mathbb{E}\left[\overline{\mathcal{H}}_{ijk}\right] = 0$. To prove the result by the non-commutative Bernstein inequality, we need to bound $\left\|\overline{\mathcal{H}}_{ijk}\right\|$ and $\left\|\sum_{i,j,k}\mathbb{E}\left[\overline{\mathcal{H}}_{i,j,k}^2\right]\right\|$. Firstly,

we have

$$
\begin{aligned}
\left\|\overline{\mathcal{H}}_{ijk}\right\| &= \sup_{\|\overline{\mathcal{X}}\|_{\mathrm{F}}=1} \left\|\overline{\mathcal{H}}_{i,j,k}(\overline{\mathcal{X}})\right\|_{\mathrm{F}} \\
&= \sup_{\|\overline{\mathcal{X}}\|_{\mathrm{F}}=1} \left\|(\frac{1}{p}\delta_{ijk}-1)\left\langle \mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^\top, \mathcal{P}_{\mathbb{T}}(\mathcal{X})\right\rangle \overline{\mathcal{P}_{\mathbb{T}}\left(\mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^\top\right)}\right\|_{\mathrm{F}} \\
&= \sup_{\|\overline{\mathcal{X}}\|_{\mathrm{F}}=1} \left\|(\frac{1}{p}\delta_{ijk}-1)\left\langle \mathcal{P}_{\mathbb{T}}(\mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^\top), \mathcal{X}\right\rangle \overline{\mathcal{P}_{\mathbb{T}}\left(\mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^\top\right)}\right\|_{\mathrm{F}} \\
&\leq \sup_{\|\overline{\mathcal{X}}\|_{\mathrm{F}}=1} \frac{1}{p}\left\|\mathcal{P}_{\mathbb{T}}(\mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^\top)\right\|_{\mathrm{F}} \|\mathcal{X}\|_{\mathrm{F}} \left\|\overline{\mathcal{P}_{\mathbb{T}}(\mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^\top)}\right\|_{\mathrm{F}} \\
&= \sup_{\|\overline{\mathcal{X}}\|_{\mathrm{F}}=1} \frac{1}{p}\left\|\mathcal{P}_{\mathbb{T}}(\mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^\top)\right\|_{\mathrm{F}} \|\mathcal{X}\|_{\mathrm{F}}\sqrt{n_3} \left\|\mathcal{P}_{\mathbb{T}}(\mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^\top)\right\|_{\mathrm{F}} \\
&= \sup_{\|\overline{\mathcal{X}}\|_{\mathrm{F}}=1} \frac{1}{p}\left\|\mathcal{P}_{\mathbb{T}}(\mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \text{`}\mathring{\mathbf{e}}_j^\top)\right\|_{\mathrm{F}} \|\overline{\mathcal{X}}\|_{\mathrm{F}} \left\|\mathcal{P}_{\mathbb{T}}(\mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^\top)\right\|_{\mathrm{F}} \\
&= \frac{1}{p}\left\|\mathcal{P}_{\mathbb{T}}(\mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^\top)\right\|_{\mathrm{F}}^2 \leq \frac{\mu_0(n_1+n_2)r}{n_1 n_2 p}.
\end{aligned}
$$

Next, we move on to bound $\left\|\sum_{i,j,k}\mathbb{E}\left[\overline{\mathcal{H}}_{i,j,k}^2\right]\right\|$. By using the fact that $\mathcal{P}_{\mathbb{T}}$ is a self-adjoint and an idempotent operator, we can get that

$$
\overline{\mathcal{H}}_{i,j,k}^2(\overline{\mathcal{X}}) = \left(\frac{1}{p}\delta_{ijk}-1\right)^2 \langle \mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^\top, \mathcal{P}_{\mathbb{T}}(\mathcal{X})\rangle\langle \mathbf{e}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^\top, \mathcal{P}_{\mathbb{T}}(\mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^\top)\rangle \overline{\mathcal{P}_{\mathbb{T}}\left(\mathbf{e}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^\top\right)}.
$$

Note that $\mathbb{E}\left[\left(\frac{1}{p}\delta_{ijk}-1\right)^2\right]=\frac{1-p}{p}\leq\frac{1}{p}$. Notice that

$$
\begin{aligned}
&\left\|\sum_{i,j,k}\mathbb{E}\left[\overline{\mathcal{H}}_{ijk}^2(\mathcal{X})\right]\right\|_{\mathrm{F}}\\
\leq&\frac{1}{p}\left\|\sum_{i,j,k}\langle\mathring{\mathbf{e}}_i*\dot{\mathbf{e}}_k*\mathring{\mathbf{e}}_j^\top,\mathcal{P}_{\mathbb{T}}(\mathcal{X})\rangle\langle\mathring{\mathbf{e}}_i*\dot{\mathbf{e}}_k*\mathring{\mathbf{e}}_j^\top,\mathcal{P}_{\mathbb{T}}\left(\mathring{\mathbf{e}}_i*\dot{\mathbf{e}}_k*\mathring{\mathbf{e}}_j^\top\right)\rangle\overline{\mathcal{P}_{\mathbb{T}}\left(\mathring{\mathbf{e}}_i*\dot{\mathbf{e}}_k*\mathring{\mathbf{e}}_j^\top\right)}\right\|_{\mathrm{F}}\\
=&\frac{\sqrt{n_3}}{p}\left\|\sum_{i,j,k}\left\langle\mathring{\mathbf{e}}_i*\dot{\mathbf{e}}_k*\mathring{\mathbf{e}}_j^\top,\mathcal{P}_{\mathbb{T}}(\mathcal{X})\right\rangle\left\langle\mathring{\mathbf{e}}_i*\dot{\mathbf{e}}_k*\mathring{\mathbf{e}}_j^\top,\mathcal{P}_{\mathbb{T}}\left(\mathring{\mathbf{e}}_i*\dot{\mathbf{e}}_k*\mathring{\mathbf{e}}_j^\top\right)\right\rangle\mathcal{P}_{\mathbb{T}}\left(\mathring{\mathbf{e}}_i*\dot{\mathbf{e}}_k*\mathring{\mathbf{e}}_j^\top\right)\right\|_{\mathrm{F}}\\
\leq&\frac{\sqrt{n_3}}{p}\left\|\sum_{i,j,k}\left\langle\mathring{\mathbf{e}}_i*\dot{\mathbf{e}}_k*\mathring{\mathbf{e}}_j^\top,\mathcal{P}_{\mathbb{T}}(\mathcal{X})\right\rangle\left\langle\mathring{\mathbf{e}}_i*\dot{\mathbf{e}}_k*\mathring{\mathbf{e}}_j^\top,\mathcal{P}_{\mathbb{T}}\left(\mathring{\mathbf{e}}_i*\dot{\mathbf{e}}_k*\mathring{\mathbf{e}}_j^\top\right)\right\rangle\cdot\left(\mathring{\mathbf{e}}_i*\dot{\mathbf{e}}_k*\mathring{\mathbf{e}}_j^\top\right)\right\|_{\mathrm{F}}\\
\leq&\frac{\sqrt{n_3}}{p}\cdot\max_{i,j,k}\left\{\left\langle\mathring{\mathbf{e}}_i*\dot{\mathbf{e}}_k*\mathring{\mathbf{e}}_j^\top,\mathcal{P}_{\mathbb{T}}\left(\mathring{\mathbf{e}}_i*\dot{\mathbf{e}}_k*\mathring{\mathbf{e}}_j^\top\right)\right\rangle\right\}\left\|\sum_{i,j,k}\left\langle\mathring{\mathbf{e}}_i*\dot{\mathbf{e}}_k*\mathring{\mathbf{e}}_j^\top,\mathcal{P}_{\mathbb{T}}(\mathcal{X})\right\rangle\cdot\left(\mathring{\mathbf{e}}_i*\dot{\mathbf{e}}_k*\mathring{\mathbf{e}}_j^\top\right)\right\|_{\mathrm{F}}\\
=&\frac{\sqrt{n_3}}{p}\cdot(\max_{i,j,k}\left\|\mathcal{P}_{\mathbb{T}}\left(\mathring{\mathbf{e}}_i*\dot{\mathbf{e}}_k*\mathring{\mathbf{e}}_j^\top\right)\right\|_{\mathrm{F}}^2)\cdot\|\mathcal{P}_{\mathbb{T}}(\mathcal{X})\|_{\mathrm{F}}\\
\leq&\frac{\sqrt{n_3}(n_1+n_2)\mu_0 r}{pn_1n_2}\|\mathcal{P}_{\mathbb{T}}(\mathcal{X})\|_{\mathrm{F}}\text{ (By Lemma 2.7)}\\
\leq&\frac{\sqrt{n_3}(n_1+n_2)\mu_0 r}{pn_1n_2}\|\mathcal{X}\|_{\mathrm{F}}=\frac{(n_1+n_2)\mu_0 r}{pn_1n_2}\|\overline{\mathcal{X}}\|_{\mathrm{F}}.
\end{aligned}
$$

We have operator norm of $\sum_{i,j,k}\mathbb{E}\left[\overline{\mathcal{H}}_{ijk}^2\right]$ is bounded by $\frac{(n_1+n_2)\mu_0 r}{pn_1n_2}$. Thus, we use non-commutative Bernstein inequality to the following result:

Notice that $\frac{\sigma^2}{M}=1$ since $\sigma^2=M=\frac{(n_1+n_2)\mu_0 r}{pn_1n_2}$. Thus, by Lemma 2.5, we have

$$
\begin{aligned}
\mathbb{P}\left[\|\mathcal{P}_{\mathbb{T}}\mathcal{P}_{\Omega}\mathcal{P}_{\mathbb{T}}-\mathcal{P}_{\mathbb{T}}\|>\frac{1}{2}\right]=\mathbb{P}\left[\left\|\sum_{i,j,k}\mathcal{H}_{ijk}\right\|>\frac{1}{2}\right]=&\mathbb{P}\left[\left\|\sum_{i,j,k}\overline{\mathcal{H}}_{ijk}\right\|>\frac{1}{2}\right]\\
\leq&2n_1n_2n_3\exp\left(-\frac{3pn_1n_2}{28(n_1+n_2)\mu_0 r}\right).
\end{aligned}
$$

□

Proof of Lemma 2.9. It is easy to check that

$$
\mathcal{R}_{\Omega}(\mathcal{X})-\mathcal{X}=\sum_{i,j,k}\left(\frac{1}{p}\delta_{i,j,k}-1\right)[\mathcal{X}]_{i,j,k}\mathring{\mathbf{e}}_i*\dot{\mathbf{e}}_k*\mathring{\mathbf{e}}_j^\top=:\sum_{i,j,k}\mathcal{E}_{i,j,k}.
$$

Notice that $\mathbb{E}[\overline{\mathcal{E}_{i,j,k}}] = 0$ and $\|\overline{\mathcal{E}_{i,j,k}}\| \leq \frac{1}{p}\|\mathcal{X}\|_\infty$. In order to use the non-commutative Bernstein inequality, we just need to check uniform boundness of the spectral norm of $\mathbb{E}(\sum_{i,j,k}(\overline{\mathcal{E}_{i,j,k}})^\top \overline{\mathcal{E}_{i,j,k}})$ and $\mathbb{E}(\sum_{i,j,k}\overline{\mathcal{E}_{i,j,k}}(\overline{\mathcal{E}_{i,j,k}})^\top)$. Using the fact that $\mathring{\mathbf{e}}_k^\top * \mathring{\mathbf{e}}_k = \dot{\mathbf{e}}_k^\top * \dot{\mathbf{e}}_k = \dot{\mathbf{e}}_1, \dot{\mathbf{e}}_1 * \dot{\mathbf{e}}_k = \dot{\mathbf{e}}_k$, and $\mathring{\mathbf{e}}_j * \dot{\mathbf{e}}_1 = \mathring{\mathbf{e}}_j$, we can have the following result:

$$
\begin{aligned}
\mathcal{E}_{i,j,k}^\top(\mathcal{T}) * \mathcal{E}_{i,j,k}(\mathcal{X}) &= \left(\frac{1}{p}\delta_{i,j,k} - 1\right)^2 [\mathcal{X}]_{i,j,k}^2 \left(\mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^\top\right)^\top * \left(\mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^\top\right) \\
&= \left(\frac{1}{p}\delta_{i,j,k} - 1\right)^2 [\mathcal{X}]_{i,j,k}^2 \left(\mathring{\mathbf{e}}_j * \dot{\mathbf{e}}_k^\top * \mathring{\mathbf{e}}_i^\top\right) * \left(\mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^\top\right) \\
&= \left(\frac{1}{p}\delta_{i,j,k} - 1\right)^2 [\mathcal{X}]_{i,j,k}^2 \mathring{\mathbf{e}}_j * \dot{\mathbf{e}}_k^\top * \left(\mathring{\mathbf{e}}_i^\top * \mathring{\mathbf{e}}_i\right) * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^\top \\
&= \left(\frac{1}{p}\delta_{i,j,k} - 1\right)^2 [\mathcal{X}]_{i,j,k}^2 \mathring{\mathbf{e}}_j * \dot{\mathbf{e}}_k^\top * (\dot{\mathbf{e}}_1 * \dot{\mathbf{e}}_k) * \mathring{\mathbf{e}}_j^\top \\
&= \left(\frac{1}{p}\delta_{i,j,k} - 1\right)^2 [\mathcal{X}]_{i,j,k}^2 \mathring{\mathbf{e}}_j * \left(\dot{\mathbf{e}}_k^\top * \dot{\mathbf{e}}_k\right) * \mathring{\mathbf{e}}_j^\top \\
&= \left(\frac{1}{p}\delta_{i,j,k} - 1\right)^2 [\mathcal{X}]_{i,j,k}^2 \mathring{\mathbf{e}}_j * (\dot{\mathbf{e}}_1) * \mathring{\mathbf{e}}_j^\top \\
&= \left(\frac{1}{p}\delta_{i,j,k} - 1\right)^2 [\mathcal{X}]_{i,j,k}^2 \mathring{\mathbf{e}}_j * \mathring{\mathbf{e}}_j^\top.
\end{aligned}
$$

Notice that $\mathring{\mathbf{e}}_j * \mathring{\mathbf{e}}_j^\top$ returns a zero tensor except for $(j,j,1)$th entry being 1. We have

$$\begin{aligned}
\left\|\sum_{i,j,k}\mathbb{E}[\overline{\mathcal{E}}_{i,j,k}^\top\overline{\mathcal{E}}_{i,j,k}]\right\| &= \left\|\sum_{i,j,k}\mathbb{E}[\mathcal{E}_{i,j,k}^\top * \mathcal{E}_{i,j,k}]\right\| \\
&\le \frac{1}{p}\left\|\sum_{i,j,k}[\mathcal{X}]_{i,j,k}^2\mathring{\mathbf{e}}_j * \mathring{\mathbf{e}}_j^\top\right\| \\
&= \frac{1}{p}\left\|\sum_{i,j,k}[\mathcal{X}]_{i,j,k}^2\overline{\mathring{\mathbf{e}}}_j\overline{\mathring{\mathbf{e}}}_j^\top\right\| \text{ (by the definition of spectral norm of tensor)} \\
&= \frac{1}{p}\left\|\begin{pmatrix}
\sum\limits_{i,k}[\mathcal{X}]_{i,1,k}^2 & & & & \\
& \sum\limits_{i,k}[\mathcal{X}]_{i,2,k}^2 & & & \\
& & \ddots & & \\
& & & \sum\limits_{i,k}[\mathcal{X}]_{i,n_2,k}^2 & \\
& & & & O_{n_2(n_3-1)\times n_2(n_3-1)}
\end{pmatrix}\right\| \\
&= \frac{1}{p}\max_{j\in[n_3]}\left\{\sum_{i,k}[\mathcal{X}]_{i,j,k}^2\right\} = \frac{1}{p}\max_{j\in[n_3]}\left\|[\mathcal{X}]_{:,j,:}\right\|_{\mathrm{F}}^2.
\end{aligned}$$

Similarly, we can get $\left\|\sum_{i,j,k}\mathbb{E}[\overline{\mathcal{E}}_{i,j,k}\overline{\mathcal{E}}_{i,j,k}^\top]\right\| \le \frac{1}{p}\max_{i\in[n_3]}\left\|[\mathcal{X}]_{i,:,:}\right\|_{\mathrm{F}}^2$. Thus,

$$\max\left\{\mathbb{E}(\sum_{i,j,k}(\overline{\mathcal{E}_{i,j,k}})^\top\overline{\mathcal{E}_{i,j,k}}), \mathbb{E}(\sum_{i,j,k}\overline{\mathcal{E}_{i,j,k}}(\overline{\mathcal{E}_{i,j,k}})^\top)\right\} \le \frac{1}{p}\left\|\mathcal{X}\right\|_{\infty,2}^2.$$

By Lemma 2.6, for any $c > 1$,

$$\begin{aligned}
\|\mathcal{R}_\Omega(\mathcal{X}) - \mathcal{X}\| &= \left\|\overline{\mathcal{R}_\Omega} - \overline{\mathcal{X}}\right\| = \left\|\sum_{i,j,k}\overline{\mathcal{E}_{i,j,k}}\right\| \\
&\le \sqrt{\frac{2c_2}{p}\|\mathcal{X}\|_{\infty,2}^2\log((n_1+n_2)n_3)} + \frac{c_2\log((n_1+n_2)n_3)}{p}\|\mathcal{T}\|_\infty
\end{aligned}$$

holds with probability at least $1 - ((n_1+n_2)n_3)^{1-c_2}$.

Proof of Lemma 2.10 Consider any b-th lateral column of $\mathcal{P}_{\mathbb{T}}\mathcal{R}_\Omega(\mathcal{X}) - \mathcal{P}_{\mathbb{T}}(\mathcal{X})$:

$$(\mathcal{P}_{\mathbb{T}}\mathcal{R}_\Omega(\mathcal{X}) - \mathcal{P}_{\mathbb{T}}(\mathcal{X})) * \mathring{\mathbf{e}}_b = \sum_{i,j,k}(\frac{1}{p}\delta_{i,j,k} - 1)[\mathcal{X}]_{i,j,k}\mathcal{P}_{\mathbb{T}}(\mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^\top) * \mathring{\mathbf{e}}_b =: \sum_{i,j,k}\mathfrak{a}_{i,j,k},$$

where $\mathfrak{a}_{i,j,k} \in \mathbb{K}^{n_1\times 1\times n_3}$ are zero-mean independent lateral tensor columns.

Denote $\vec{\mathfrak{a}}_{i,j,k} \in \mathbb{K}^{n_1 n_3 \times 1}$ as the vectorized column vector of $\mathfrak{a}_{i,j,k}$. Then, we have

$$\left\|\vec{\mathfrak{a}}_{i,j,k}\right\| = \left\|\mathfrak{a}_{i,j,k}\right\|_{\mathrm{F}} \leq \frac{1}{p}|[\mathcal{X}]_{i,j,k}| \left\|\mathcal{P}_{\mathbb{T}}(\mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^\top) * \mathring{\mathbf{e}}_b\right\|_{\mathrm{F}} \leq \frac{1}{p}\sqrt{\frac{(n_1+n_2)\mu_0 r}{n_1 n_2}}\|\mathcal{X}\|_\infty.$$

We also have

$$\left|\mathbb{E}(\sum_{i,j,k} \vec{\mathfrak{a}}_{i,j,k}^\top \vec{\mathfrak{a}}_{i,j,k})\right| = \mathbb{E}(\sum_{i,j,k} \left\|\mathfrak{a}_{i,j,k}\right\|_{\mathrm{F}}^2) = \frac{1-p}{p}\sum_{i,j,k}[\mathcal{X}]_{i,j,k}^2 \left\|\mathcal{P}_{\mathbb{T}}(\mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^\top) * \mathring{\mathbf{e}}_b\right\|_{\mathrm{F}}^2.$$

By the definition of $\mathcal{P}_{\mathbb{T}}$ and the incoherence condition, we have:

$$\begin{aligned}
&\left\|\mathcal{P}_{\mathbb{T}}(\mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^\top) * \mathring{\mathbf{e}}_b)\right\|_{\mathrm{F}} \\
&= \left\|(\mathcal{U} * \mathcal{U}^\top * \mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k) * \mathring{\mathbf{e}}_j^\top * \mathring{\mathbf{e}}_b + (\mathcal{I}_{n_1} - \mathcal{U} * \mathcal{U}^\top) * \mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^\top * \mathcal{V} * \mathcal{V}^\top * \mathring{\mathbf{e}}_b\right\|_{\mathrm{F}} \\
&\leq \sqrt{\frac{\mu_0 r}{n_1}}\left\|\mathring{\mathbf{e}}_j^\top * \mathring{\mathbf{e}}_b\right\|_{\mathrm{F}} + \left\|(\mathcal{I}_{n_1} - \mathcal{U} * \mathcal{U}^\top) * \mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k\right\|\left\|\mathring{\mathbf{e}}_j^\top * \mathcal{V} * \mathcal{V}^\top * \mathring{\mathbf{e}}_b\right\|_{\mathrm{F}} \\
&\leq \sqrt{\frac{\mu_0 r}{n_1}}\left\|\mathring{\mathbf{e}}_j^\top * \mathring{\mathbf{e}}_b\right\|_{\mathrm{F}} + \left\|\mathring{\mathbf{e}}_j^\top * \mathcal{V} * \mathcal{V}^\top * \mathring{\mathbf{e}}_b\right\|_{\mathrm{F}}
\end{aligned}$$

By Cauchy-Schwartz inequality, we have

$$\left\|\mathcal{P}_{\mathbb{T}}(\mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^\top) * \mathring{\mathbf{e}}_b)\right\|_{\mathrm{F}}^2 \leq \frac{2\mu_0 r}{n_1}\left\|\mathring{\mathbf{e}}_j^\top * \mathring{\mathbf{e}}_b\right\|_{\mathrm{F}}^2 + 2\left\|\mathring{\mathbf{e}}_j^\top * \mathcal{V} * \mathcal{V}^\top * \mathring{\mathbf{e}}_b\right\|_{\mathrm{F}}^2.$$

Thus,

$$\begin{aligned}
\left|\mathbb{E}(\sum_{i,j,k} \vec{\mathfrak{a}}_{i,j,k}^\top \vec{\mathfrak{a}}_{i,j,k})\right| &\leq \frac{2\mu_0 r}{pn_1}\sum_{i,j,k}[\mathcal{X}]_{i,j,k}^2\left\|\mathring{\mathbf{e}}_j^\top * \mathring{\mathbf{e}}_b^\circ\right\|_{\mathrm{F}}^2 + \frac{2}{p}\sum_{i,j,k}[\mathcal{X}]_{i,j,k}^2\left\|\mathring{\mathbf{e}}_j^\top * \mathcal{V} * \mathcal{V}^\top * \mathring{\mathbf{e}}_b^\circ\right\|_{\mathrm{F}}^2 \\
&= \frac{2\mu_0 r}{pn_1}\sum_{i,j,k}[\mathcal{X}]_{i,j,k}^2\left\|\mathring{\mathbf{e}}_j^\top * \mathring{\mathbf{e}}_b^\circ\right\|_{\mathrm{F}}^2 + \frac{2}{p}\sum_{j}\left\|\mathring{\mathbf{e}}_j^\top * \mathcal{V} * \mathcal{V}^\top * \mathring{\mathbf{e}}_b^\circ\right\|_{\mathrm{F}}^2\sum_{i,k}[\mathcal{X}]_{i,j,k}^2 \\
&\leq \frac{2\mu_0 r}{pn_1}\sum_{i,j,k}[\mathcal{X}]_{i,j,k}^2\left\|\mathring{\mathbf{e}}_j^\top * \mathring{\mathbf{e}}_b^\circ\right\|_{\mathrm{F}}^2 + \frac{2}{p}\sum_{j}\left\|\mathring{\mathbf{e}}_j^\top * \mathcal{V} * \mathcal{V}^\top * \mathring{\mathbf{e}}_b^\circ\right\|_{\mathrm{F}}^2\|\mathcal{X}\|_{\infty,2}^2 \\
&= \frac{2\mu_0 r}{pn_1}\sum_{i,k}[\mathcal{X}]_{i,b,k}^2 + \frac{2}{p}\left\|\mathcal{V} * \mathcal{V}^\top * \mathring{\mathbf{e}}_b^\circ\right\|_{\mathrm{F}}^2\|\mathcal{X}\|_{\infty,2}^2 \\
&\leq \frac{2\mu_0 r}{pn_1}\|\mathcal{X}\|_{\infty,2}^2 + \frac{2\mu_0 r}{pn_2}\|\mathcal{X}\|_{\infty,2}^2 \leq \frac{2(n_1+n_2)\mu_0 r}{pn_1 n_2}\|\mathcal{X}\|_{\infty,2}^2
\end{aligned}$$

Similarly, we can bound $\left|\mathbb{E}(\sum_{i,j,k} \vec{\mathfrak{a}}_{i,j,k}\vec{\mathfrak{a}}_{i,j,k}^\top)\right|$ by the same quantity.

For simplicity, we let $M = \frac{1}{p}\sqrt{\frac{(n_1+n_2)\mu_0 r}{n_1 n_2}}\|\mathcal{X}\|_\infty$ and $\sigma^2 = \frac{2(n_1+n_2)\mu_0 r}{pn_1 n_2}\|\mathcal{X}\|_{\infty,2}^2$.

By Lemma 2.6, for any $c_1 > 1$, we have

$$\mathbb{P}\left(\left\|\sum_{i,j,k}\vec{\mathfrak{a}}_{i,j,k}\right\| \le \sqrt{2c_1\sigma^2\log((n_1+n_2)n_3)} + c_1 M\log((n_1+n_2)n_3)\right)$$

$$= \mathbb{P}\left(\left\|\sum_{i,j,k}\vec{\mathfrak{a}}_{i,j,k}\right\| \le \sqrt{\frac{4c_1\log\left((n_1+n_2)n_3\right)(n_1+n_2)\mu_0 r}{pn_1n_2}}\cdot\|\mathcal{X}\|_{\infty,2}\right.$$

$$\left. + \frac{c_1\log((n_1+n_2)n_3)\sqrt{(n_1+n_2)\mu_0 r}}{p\sqrt{n_1n_2}}\|\mathcal{X}\|_\infty\right) \ge 1-((n_1+n_2)n_3)^{1-c_1}.$$

Notice that

$$\|(\mathcal{P}_{\mathbb{T}}\mathcal{R}_\Omega(\mathcal{X})-\mathcal{P}_{\mathbb{T}}(\mathcal{X}))*\mathring{\mathbf{e}}_b\|_{\mathrm{F}} = \left\|\sum_{i,j,k}\mathfrak{a}_{i,j,k}\right\|_{\mathrm{F}} = \left\|\sum_{i,j,k}\vec{\mathfrak{a}}_{i,j,k}\right\|.$$

Therefore,

$$\mathbb{P}\left(\|(\mathcal{P}_{\mathbb{T}}\mathcal{R}_\Omega(\mathcal{X})-\mathcal{P}_{\mathbb{T}}(\mathcal{X}))*\mathring{\mathbf{e}}_b\|_{\mathrm{F}} \le \sqrt{\frac{4c_1(n_1+n_2)\log\left((n_1+n_2)n_3\right)\mu_0 r}{pn_1n_2}}\cdot\|\mathcal{X}\|_{\infty,2}+\right.$$

$$\left.\frac{c_1\log((n_1+n_2)n_3)\sqrt{(n_1+n_2)\mu_0 r}}{p\sqrt{n_1n_2}}\|\mathcal{X}\|_\infty\right) \ge 1-((n_1+n_2)n_3)^{1-c_1}.$$

Using a union bound over all the tensor lateral slices, we have

$$\mathbb{P}\left(\max_b\left\{\|(\mathcal{P}_{\mathbb{T}}\mathcal{R}_\Omega(\mathcal{X})-\mathcal{P}_{\mathbb{T}}(\mathcal{X}))*\mathring{\mathbf{e}}_b\|_{\mathrm{F}}\right\} \le \sqrt{\frac{4c_1(n_1+n_2)\log\left((n_1+n_2)n_3\right)\mu_0 r}{pn_1n_2}}\cdot\|\mathcal{X}\|_{\infty,2}\right.$$

$$\left.+\frac{c_1\log((n_1+n_2)n_3)\sqrt{(n_1+n_2)\mu_0 r}}{p\sqrt{n_1n_2}}\|\mathcal{X}\|_\infty\right) \ge 1-n_2((n_1+n_2)n_3)^{1-c_1}.$$

Similarly, we can also show that

$$\mathbb{P}\left(\max_b\left\{\left\|\mathring{\mathbf{e}}_b^\top*(\mathcal{P}_{\mathbb{T}}\mathcal{R}_\Omega(\mathcal{X})-\mathcal{P}_{\mathbb{T}}(\mathcal{X}))\right\|_{\mathrm{F}}\right\} \le \sqrt{\frac{4c_1(n_1+n_2)\log\left((n_1+n_2)n_3\right)\mu_0 r}{pn_1n_2}}\cdot\|\mathcal{X}\|_{\infty,2}+\right.$$

$$\left.\frac{c_1\log((n_1+n_2)n_3)\sqrt{(n_1+n_2)\mu_0 r}}{p\sqrt{n_1n_2}}\|\mathcal{X}\|_\infty\right) \ge 1-n_1((n_1+n_2)n_3)^{1-c_1}.$$

Thus, we can get

$$\mathbb{P}\left(\|(\mathcal{P}_{\mathbb{T}}\mathcal{R}_\Omega(\mathcal{X})-\mathcal{P}_{\mathbb{T}}(\mathcal{X}))\|_{\infty,2} \le \sqrt{\frac{4c_1(n_1+n_2)\log\left((n_1+n_2)n_3\right)\mu_0 r}{pn_1n_2}}\cdot\|\mathcal{X}\|_{\infty,2}\right.$$

$$\left.+\frac{c_1\log((n_1+n_2)n_3)\sqrt{(n_1+n_2)\mu_0 r}}{p\sqrt{n_1n_2}}\|\mathcal{X}\|_\infty\right) \ge 1-((n_1+n_2)n_3)^{2-c_1}.$$

□

Proof of Lemma 2.11. Notice that

$$\mathcal{P}_{\mathbb{T}}\mathcal{R}_{\Omega}\mathcal{P}_{\mathbb{T}}(\mathcal{X}) = \mathcal{P}_{\mathbb{T}}\mathcal{R}_{\Omega}\left(\sum_{i,j,k}\left\langle \mathcal{P}_{\mathbb{T}}(\mathcal{X}), \mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^{\top}\right\rangle \mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^{\top}\right) \tag{2.35}$$

$$= \mathcal{P}_{\mathbb{T}}\left(\sum_{i,j,k}\frac{1}{p}\delta_{i,j,k}\left\langle \mathcal{P}_{\mathbb{T}}(\mathcal{X}), \mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^{\top}\right\rangle \mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^{\top}\right)$$

$$= \sum_{i,j,k}\frac{1}{p}\delta_{i,j,k}\left\langle \mathcal{P}_{\mathbb{T}}(\mathcal{X}), \mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^{\top}\right\rangle \mathcal{P}_{\mathbb{T}}\left(\mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^{\top}\right). \tag{2.36}$$

Equation (2.35) is due to $\mathcal{P}_{\mathbb{T}}(\mathcal{X}) = \sum_{i,j,k}\left\langle \mathcal{P}_{\mathbb{T}}(\mathcal{X}), \mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^{\top}\right\rangle \mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^{\top}$. Equation (2.36) is due to linearity of operator $\mathcal{P}_{\mathbb{T}}$.

Notice that

$$\mathcal{P}_{\mathbb{T}}(\mathcal{X}) = \mathcal{P}_{\mathbb{T}}\left(\mathcal{P}_{\mathbb{T}}(\mathcal{X})\right)$$

$$= \mathcal{P}_{\mathbb{T}}\left(\sum_{i,j,k}\left\langle \mathcal{P}_{\mathbb{T}}(\mathcal{X}), \mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^{\top}\right\rangle \mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^{\top}\right)$$

$$= \sum_{i,j,k}\left\langle \mathcal{P}_{\mathbb{T}}(\mathcal{X}), \mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^{\top}\right\rangle \mathcal{P}_{\mathbb{T}}\left(\mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^{\top}\right)$$

Thus, we can have any (a, b, c)th entry of $\mathcal{P}_{\mathbb{T}}\mathcal{R}_{\Omega}\mathcal{P}_{\mathbb{T}}(\mathcal{X}) - \mathcal{P}_{\mathbb{T}}(\mathcal{X})$ can be given by

$$\left\langle \mathcal{P}_{\mathbb{T}}\mathcal{R}_{\Omega}\mathcal{P}_{\mathbb{T}}(\mathcal{X}) - \mathcal{P}_{\mathbb{T}}(\mathcal{X}), \mathring{\mathbf{e}}_a * \dot{\mathbf{e}}_c * \mathring{\mathbf{e}}_b^{\top}\right\rangle$$

$$= \left\langle \sum_{i,j,k}(\frac{1}{p}\delta_{i,j,k} - 1)\left\langle \mathcal{P}_{\mathbb{T}}(\mathcal{X}), \mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^{\top}\right\rangle \mathcal{P}_{\mathbb{T}}\left(\mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^{\top}\right), \mathring{\mathbf{e}}_a * \dot{\mathbf{e}}_c * \mathring{\mathbf{e}}_b^{\top}\right\rangle$$

$$= \sum_{i,j,k}(\frac{1}{p}\delta_{i,j,k} - 1)\left\langle \mathcal{P}_{\mathbb{T}}(\mathcal{X}), \mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^{\top}\right\rangle \left\langle \mathcal{P}_{\mathbb{T}}\left(\mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^{\top}\right), \mathring{\mathbf{e}}_a * \dot{\mathbf{e}}_c * \mathring{\mathbf{e}}_b^{\top}\right\rangle$$

$$=: \sum_{i,j,k} h_{i,j,k}.$$

It is easy to see that $\mathbb{E}(h_{i,j,k}) = 0$. Notice that

$$|h_{i,j,k}| = \left|(\frac{1}{p}\delta_{i,j,k} - 1)\left\langle \mathcal{P}_{\mathbb{T}}(\mathcal{X}), \mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^{\top}\right\rangle \left\langle \mathcal{P}_{\mathbb{T}}\left(\mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^{\top}\right), \mathring{\mathbf{e}}_a * \dot{\mathbf{e}}_c * \mathring{\mathbf{e}}_b^{\top}\right\rangle\right|$$

$$= \left|(\frac{1}{p}\delta_{i,j,k} - 1)\left\langle \mathcal{P}_{\mathbb{T}}(\mathcal{X}), \mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^{\top}\right\rangle \left\langle \mathcal{P}_{\mathbb{T}}\left(\mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^{\top}\right), \mathcal{P}_{\mathbb{T}}\left(\mathring{\mathbf{e}}_a * \dot{\mathbf{e}}_c * \mathring{\mathbf{e}}_b^{\top}\right)\right\rangle\right|$$

$$\leq \frac{1}{p}\|\mathcal{P}_{\mathbb{T}}(\mathcal{X})\|_{\infty}\left\|\mathcal{P}_{\mathbb{T}}\left(\mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^{\top}\right)\right\|_{\mathrm{F}}\left\|\mathcal{P}_{\mathbb{T}}\left(\mathring{\mathbf{e}}_a * \dot{\mathbf{e}}_c * \mathring{\mathbf{e}}_b^{\top}\right)\right\|_{\mathrm{F}} \leq \frac{(n_1+n_2)\mu_0 r}{pn_1n_2}\|\mathcal{P}_{\mathbb{T}}(\mathcal{X})\|_{\infty}$$

It is easy to check that

$$\left|\sum_{i,j,k}\mathbb{E}[h_{i,j,k}^2]\right| = \mathbb{E}\left(\left|(\frac{1}{p}\delta_{i,j,k}-1)\left\langle\mathcal{P}_{\mathbb{T}}(\mathcal{X}),\mathring{\mathbf{e}}_i*\dot{\mathbf{e}}_k*\mathring{\mathbf{e}}_j^\top\right\rangle\left\langle\mathcal{P}_{\mathbb{T}}\left(\mathring{\mathbf{e}}_i*\dot{\mathbf{e}}_k*\mathring{\mathbf{e}}_j^\top\right),\mathring{\mathbf{e}}_a*\dot{\mathbf{e}}_c*\mathring{\mathbf{e}}_b^\top\right\rangle\right|^2\right)$$
$$= \mathbb{E}\left((\frac{1}{p}\delta_{i,j,k}-1)^2\left|\left\langle\mathcal{P}_{\mathbb{T}}(\mathcal{X}),\mathring{\mathbf{e}}_i*\dot{\mathbf{e}}_k*\mathring{\mathbf{e}}_j^\top\right\rangle\left\langle\mathcal{P}_{\mathbb{T}}\left(\mathring{\mathbf{e}}_i*\dot{\mathbf{e}}_k*\mathring{\mathbf{e}}_j^\top\right),\mathring{\mathbf{e}}_a*\dot{\mathbf{e}}_c*\mathring{\mathbf{e}}_b^\top\right\rangle\right|^2\right)$$
$$\leq \frac{1}{p}\left\|\mathcal{P}_{\mathbb{T}}(\mathcal{X})\right\|_\infty^2\left\|\mathcal{P}_{\mathbb{T}}\left(\mathring{\mathbf{e}}_i*\dot{\mathbf{e}}_k*\mathring{\mathbf{e}}_j^\top\right)\right\|_{\mathrm{F}}^2 \leq \frac{(n_1+n_2)\mu_0 r}{pn_1n_2}\left\|\mathcal{P}_{\mathbb{T}}(\mathcal{X})\right\|_\infty^2.$$

Thus, by non-commutative Bernstein inequality, we have

$$\mathbb{P}\left[\left(\mathcal{P}_{\mathbb{T}}\mathcal{R}_\Omega\mathcal{P}_{\mathbb{T}}(\mathcal{X})-\mathcal{P}_{\mathbb{T}}(\mathcal{X})\right)_{a,b,c}\geq\frac{1}{2}\|\mathcal{P}_{\mathbb{T}}(\mathcal{X})\|_\infty\right]$$
$$\leq 2\exp\left(\frac{-\|\mathcal{P}_{\mathbb{T}}(\mathcal{X})\|_\infty^2/8}{\frac{(n_1+n_2)\mu_0 r}{pn_1n_2}\|\mathcal{P}_{\mathbb{T}}(\mathcal{X})\|_\infty^2+\frac{(n_1+n_2)\mu_0 r}{6pn_1n_2}\|\mathcal{P}_{\mathbb{T}}(\mathcal{X})\|_\infty^2}\right)$$
$$= 2\exp\left(\frac{-3pn_1n_2}{28(n_1+n_2)\mu_0 r}\right).$$

Thus, using the union bound on every (a,b,c)th entry, we have

$$\left\|\left(\mathcal{P}_{\mathbb{T}}\mathcal{R}_\Omega\mathcal{P}_{\mathbb{T}}-\mathcal{P}_{\mathbb{T}}\right)\left(\mathcal{P}_{\mathbb{T}}(\mathcal{X})\right)\right\|_\infty\leq\frac{1}{2}\|\mathcal{P}_{\mathbb{T}}(\mathcal{X})\|_\infty$$

holds with probability at least $1-2n_1n_2n_3\exp\left(\frac{-3pn_1n_2}{28(n_1+n_2)\mu_0 r}\right)$.

Lastly, to maintain the integrity of a self-contained exposition, we offer a detailed proof of Proposition 1 in Section 2.6.2.3, as originally presented in [84]. □

Proof of Proposition 1 In the following context, the symbol $\mathcal{T}$ is used to represent the tensor that we aim to recover in the optimization problem (2.18). Before we delve into the detailed proof pipeline, we wish to reiterate the purpose of Proposition 1. It asserts that $\mathcal{T}$ is a unique minimizer to the optimization problem (2.18) when Conditions 1 and 2 are met simultaneously.

Notice that $\mathcal{T}$ is a feasible solution to the problem (2.18). In order to show $\mathcal{T}$ is the unique minimizer, it suffices to show

$$\|\mathcal{X}\|_{\mathrm{TNN}}-\|\mathcal{T}\|_{\mathrm{TNN}}>0$$

for any feasible solution $\mathcal{X}$ but $\mathcal{X}\neq\mathcal{T}$.

We first show that for any feasible solution $\mathcal{X}$ different from $\mathcal{T}$, there exists a tensor $\mathcal{M}$ such that

$$\|\mathcal{X}\|_{\mathrm{TNN}}-\|\mathcal{T}\|_{\mathrm{TNN}}\geq\left\langle\mathcal{U}*\mathcal{V}^\top+\mathcal{P}_{\mathbb{T}^\perp(\mathcal{T})}(\mathcal{M}),\mathcal{X}-\mathcal{T}\right\rangle.$$

In this way, we can transform proving $\|\mathcal{X}\|_{\text{TNN}} - \|\mathcal{T}\|_{\text{TNN}} > 0$ into showing

$$\left\langle \mathcal{U} * \mathcal{V}^\top + \mathcal{P}_{\mathbb{T}^\perp(\mathcal{T})}(\mathcal{M}), \mathcal{X} - \mathcal{T} \right\rangle > 0.$$

To prove $\langle \mathcal{U} * \mathcal{V}^\top + \mathcal{P}_{\mathbb{T}^\perp}(\mathcal{M}), \mathcal{X} - \mathcal{T} \rangle > 0$, we split

$$\left\langle \mathcal{U} * \mathcal{V}^\top + \mathcal{P}_{\mathbb{T}^\perp}(\mathcal{M}), \mathcal{X} - \mathcal{T} \right\rangle$$

into two parts

$$\langle \mathcal{P}_{\mathbb{T}^\perp}(\mathcal{M}), \mathcal{X} - \mathcal{T} \rangle \text{ and } \left\langle \mathcal{U} * \mathcal{V}^\top, \mathcal{X} - \mathcal{T} \right\rangle.$$

By the construction of $\mathcal{M}$, we can show that

$$\langle \mathcal{P}_{\mathbb{T}^\perp}(\mathcal{M}), \mathcal{X} - \mathcal{T} \rangle = \|\mathcal{P}_{\mathbb{T}^\perp}(\mathcal{X} - \mathcal{T})\|_{\text{TNN}}.$$

As for the part $\langle \mathcal{U} * \mathcal{V}^\top, \mathcal{X} - \mathcal{T} \rangle$, we need to further split it into two parts by introducing the dual certification tensor $\mathcal{Y}$:

$$\left\langle \mathcal{P}_{\mathbb{T}(\mathcal{T})}(\mathcal{Y}) - \mathcal{U} * \mathcal{V}^\top, \mathcal{X} - \mathcal{T} \right\rangle, \left\langle \mathcal{P}_{\mathbb{T}^\perp(\mathcal{T})}(\mathcal{Y}), \mathcal{X} - \mathcal{T} \right\rangle.$$

The reason of doing such separation is that we can bound these two terms by $\frac{1}{2}\|\mathcal{P}_{\mathbb{T}^\perp}(\mathcal{X} - \mathcal{T})\|_{\text{TNN}}$ and $\frac{\sqrt{2}}{4}\|\mathcal{P}_{\mathbb{T}^\perp}(\mathcal{X} - \mathcal{T})\|_{\text{TNN}}$ respectively. By combining the bound of above three separations together, we can get

$$\langle \mathcal{U} * \mathcal{V}^\top + \mathcal{P}_{\mathbb{T}^\perp}(\mathcal{M}), \mathcal{X} - \mathcal{T} \rangle \geq \frac{1}{8} \|\mathcal{P}_{\mathbb{T}^\perp}(\mathcal{X} - \mathcal{T})\|_{\text{TNN}}.$$

In the end, we prove $\left\|\mathcal{P}_{\mathbb{T}^\perp(\mathcal{T})}(\mathcal{X} - \mathcal{T})\right\|_{\text{TNN}}$ strictly larger than zero by contradiction. Before we move on to the detailed proof, we will give several useful lemmas which are key to the proof of Proposition 1. First, we state the characterization of the tensor nuclear norm (TNN) which can be described as a duality to the tensor spectral norm.

Lemma 2.12. *([84]) Given a tensor $\mathcal{T} \in \mathbb{K}^{n_1 \times n_2 \times n_3}$, we have*

$$\|\mathcal{T}\|_{\text{TNN}} = \sup_{\{\mathcal{Q} \in \mathbb{K}^{n_1 \times n_2 \times n_3} : \|\mathcal{Q}\| \leq 1\}} \langle \mathcal{Q}, \mathcal{T} \rangle.$$

Next, we present a characterization of the subdifferential of TNN, which is useful for proving the uniqueness of the minimizer to the optimization problem (2.18).

Lemma 2.13 (Subdifferential of TNN [84])**.** *Let $\mathcal{T} \in \mathbb{K}^{n_1 \times n_2 \times n_3}$ and its compact t-SVD be $\mathcal{T} = \mathcal{U} * \Sigma * \mathcal{V}^\top$. The subdifferential (the set of subgradients) of $\|\mathcal{T}\|_{\text{TNN}}$ is $\partial \|\mathcal{T}\|_{\text{TNN}} = \{\mathcal{U} * \mathcal{V}^\top + \mathcal{W} | \mathcal{U}^\top * \mathcal{W} = 0, \mathcal{W} * \mathcal{V} = 0, \|\mathcal{W}\| \leq 1\}$.*

For the proof of Proposition 1, a significant challenge is proving the minimizer's uniqueness. This involves ensuring the expression

$$\left\langle \mathcal{U} * \mathcal{V}^\top + \mathcal{P}_{\mathbb{T}^\perp(\mathcal{T})}(\mathcal{M}), \mathcal{X} - \mathcal{T} \right\rangle > 0$$

with $\mathcal{M}$ satisfying some conditions.

Lemma 2.14 ([84]). *Assume that Ω is generated according to the Bernoulli sampling with probability p. If $\|\mathcal{P}_{\mathbb{T}}\mathcal{R}_\Omega\mathcal{P}_{\mathbb{T}} - \mathcal{P}_{\mathbb{T}}\| \le \frac{1}{2}$, then*

$$\|\mathcal{P}_{\mathbb{T}}(\mathcal{X})\|_{\mathrm{F}} \le \sqrt{\frac{2n_3}{p}} \cdot \|\mathcal{P}_{\mathbb{T}^\perp}(\mathcal{X})\|_{\mathrm{TNN}},$$

for any $\mathcal{X}$ with $\mathcal{P}_\Omega(\mathcal{X}) = 0$.

Proof of Lemma 2.14. Let $\mathcal{X}$ be a tensor satisfying $\mathcal{P}_\Omega(\mathcal{X}) = 0$. Using self-adjoint property of the operator $\mathcal{P}_{\mathbb{T}}$, we can have

$$\begin{aligned}
\|\mathcal{R}_\Omega\mathcal{P}_{\mathbb{T}}(\mathcal{X})\|_{\mathrm{F}}^2 &= \langle \mathcal{R}_\Omega\mathcal{P}_{\mathbb{T}}(\mathcal{X}), \mathcal{R}_\Omega\mathcal{P}_{\mathbb{T}}(\mathcal{X})\rangle \\
&= \left\langle \mathcal{R}_\Omega\mathcal{P}_{\mathbb{T}}(\mathcal{X}), \sum_{i,j,k} \frac{1}{p}\delta_{i,j,k}[\mathcal{P}_{\mathbb{T}}(\mathcal{X})]_{i,j,k} \cdot \mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^\top \right\rangle \\
&= \frac{1}{p}\left\langle \mathcal{R}_\Omega\mathcal{P}_{\mathbb{T}}(\mathcal{X}), \sum_{i,j,k} \delta_{i,j,k}[\mathcal{P}_{\mathbb{T}}(\mathcal{X})]_{i,j,k} \cdot \mathring{\mathbf{e}}_i * \dot{\mathbf{e}}_k * \mathring{\mathbf{e}}_j^\top \right\rangle \\
&= \frac{1}{p}\langle \mathcal{R}_\Omega\mathcal{P}_{\mathbb{T}}(\mathcal{X}), \mathcal{P}_\Omega(\mathcal{P}_{\mathbb{T}}(\mathcal{X}))\rangle = \frac{1}{p}\langle \mathcal{P}_{\mathbb{T}}\mathcal{R}_\Omega\mathcal{P}_{\mathbb{T}}(\mathcal{X}), \mathcal{X}\rangle \\
&= \frac{1}{p}\langle \mathcal{P}_{\mathbb{T}}\mathcal{R}_\Omega\mathcal{P}_{\mathbb{T}}(\mathcal{X}) - \mathcal{P}_{\mathbb{T}}(\mathcal{X}), \mathcal{X}\rangle + \frac{1}{p}\langle \mathcal{P}_{\mathbb{T}}(\mathcal{X}), \mathcal{X}\rangle \\
&= \frac{1}{p}\|\mathcal{P}_{\mathbb{T}}(\mathcal{X})\|_{\mathrm{F}}^2 + \frac{1}{p}\langle (\mathcal{P}_{\mathbb{T}}\mathcal{R}_\Omega\mathcal{P}_{\mathbb{T}} - \mathcal{P}_{\mathbb{T}})(\mathcal{X}), \mathcal{P}_{\mathbb{T}}(\mathcal{X})\rangle \\
&\ge \frac{1}{p}\|\mathcal{P}_{\mathbb{T}}(\mathcal{X})\|_{\mathrm{F}}^2 - \frac{1}{p}\|\mathcal{P}_{\mathbb{T}}\mathcal{R}_\Omega\mathcal{P}_{\mathbb{T}} - \mathcal{P}_{\mathbb{T}}\|\langle \mathcal{X}, \mathcal{P}_{\mathbb{T}}(\mathcal{X})\rangle \\
&\ge \frac{1}{p}\|\mathcal{P}_{\mathbb{T}}(\mathcal{X})\|_{\mathrm{F}}^2 - \frac{1}{p}\|\mathcal{P}_{\mathbb{T}}\mathcal{R}_\Omega\mathcal{P}_{\mathbb{T}} - \mathcal{P}_{\mathbb{T}}\| \cdot \|\mathcal{P}_{\mathbb{T}}(\mathcal{X})\|_{\mathrm{F}}^2 \ge \frac{1}{2p}\|\mathcal{P}_{\mathbb{T}}(\mathcal{X})\|_{\mathrm{F}}^2.
\end{aligned}$$

Notice that if $\mathcal{P}_\Omega(\mathcal{X}) = 0$, then $\mathcal{R}_\Omega(\mathcal{X})$ must be zero tensor. Thus, we have $\|\mathcal{R}_\Omega\mathcal{P}_{\mathbb{T}}(\mathcal{X})\|_{\mathrm{F}} = \|\mathcal{R}_\Omega\mathcal{P}_{\mathbb{T}^\perp}(\mathcal{X})\|_{\mathrm{F}} \le \frac{1}{p}\|\mathcal{P}_{\mathbb{T}^\perp}(\mathcal{X})\|_{\mathrm{F}} = \frac{1}{p\sqrt{n_3}}\left\|\overline{\mathcal{P}_{\mathbb{T}^\perp}(\mathcal{X})}\right\|_{\mathrm{F}} \le \frac{1}{p\sqrt{n_3}}\left\|\overline{\mathcal{P}_{\mathbb{T}^\perp}(\mathcal{X})}\right\|_{*} = \frac{\sqrt{n_3}}{p}\|\mathcal{P}_{\mathbb{T}^\perp}(\mathcal{X})\|_{\mathrm{TNN}}$. As a result, we have $\|\mathcal{P}_{\mathbb{T}}(\mathcal{X})\|_{\mathrm{F}} \le \sqrt{2p}\,\|\mathcal{R}_\Omega\mathcal{P}_{\mathbb{T}}(\mathcal{X})\|_{\mathrm{F}} \le \sqrt{\frac{2n_3}{p}}\,\|\mathcal{P}_{\mathbb{T}^\perp}(\mathcal{X})\|_{\mathrm{TNN}}$. □

Proof of Proposition 1. Consider any feasible solution $\mathcal{X} \neq \mathcal{T}$ to the optimization problem (2.18) with $\mathcal{P}_\Omega(\mathcal{X}) = \mathcal{P}_\Omega(\mathcal{T})$. By the duality between the TNN and tensor spectral norm shown in

Lemma 2.12, we have that there exists a tensor $\mathcal{M}$ with $\|\mathcal{M}\| \leq 1$ such that

$$\begin{aligned}\|\mathcal{P}_{\mathbb{T}^\perp}(\mathcal{X}-\mathcal{T})\|_{\mathrm{TNN}} &= \langle \mathcal{M}, \mathcal{P}_{\mathbb{T}^\perp}(\mathcal{X}-\mathcal{T})\rangle \\ &= \langle \mathcal{P}_{\mathbb{T}^\perp}(\mathcal{M}), \mathcal{P}_{\mathbb{T}^\perp}(\mathcal{X}-\mathcal{T})\rangle .\end{aligned}$$

Firstly, it is easy to check that $\mathcal{U}^\top * \mathcal{P}_{\mathbb{T}^\perp}(\mathcal{M}) = 0$ and $\mathcal{P}_{\mathbb{T}^\perp}(\mathcal{M}) * \mathcal{V} = 0$ by the definition of the operator $\mathcal{P}_{\mathbb{T}^\perp}(\cdot)$. By Lemma 2.13, we have that $\mathcal{U} * \mathcal{V}^\top + \mathcal{P}_{\mathbb{T}^\perp}(\mathcal{M})$ is a subgradient of $\mathcal{T}$ in terms of Tensor nuclear norm. Therefore, we have

$$\|\mathcal{X}\|_{\mathrm{TNN}} - \|\mathcal{T}\|_{\mathrm{TNN}} \geq \left\langle \mathcal{U} * \mathcal{V}^\top + \mathcal{P}_{\mathbb{T}^\perp}(\mathcal{M}), \mathcal{X}-\mathcal{T}\right\rangle .$$

To prove $\|\mathcal{X}\|_{\mathrm{TNN}} - \|\mathcal{T}\|_{\mathrm{TNN}} \geq 0$, it is sufficient to show

$$\left\langle \mathcal{U} * \mathcal{V}^\top + \mathcal{P}_{\mathbb{T}^\perp}(\mathcal{M}), \mathcal{X}-\mathcal{T}\right\rangle \geq 0.$$

Notice that for any $\mathcal{Y}$ with $\mathcal{P}_\Omega(\mathcal{Y}) = \mathcal{Y}$, we have

$$\langle \mathcal{Y}, \mathcal{X}-\mathcal{T}\rangle = \langle \mathcal{P}_\Omega(\mathcal{Y}), \mathcal{X}-\mathcal{T}\rangle = \langle \mathcal{P}_\Omega(\mathcal{Y}), \mathcal{P}_\Omega(\mathcal{X}-\mathcal{T})\rangle = 0.$$

We thus have

$$\left\langle \mathcal{U} * \mathcal{V}^\top + \mathcal{P}_{\mathbb{T}^\perp}(\mathcal{M}), \mathcal{X}-\mathcal{T}\right\rangle = \left\langle \mathcal{U} * \mathcal{V}^\top + \mathcal{P}_{\mathbb{T}^\perp}(\mathcal{M}) - \mathcal{Y}, \mathcal{X}-\mathcal{T}\right\rangle .$$

Furthermore, we have

$$\begin{aligned}&\left\langle \mathcal{U} * \mathcal{V}^\top + \mathcal{P}_{\mathbb{T}^\perp}(\mathcal{M}) - \mathcal{Y}, \mathcal{X}-\mathcal{T}\right\rangle \\ =\ &\left\langle \mathcal{U} * \mathcal{V}^\top + \mathcal{P}_{\mathbb{T}^\perp}(\mathcal{M}) - \mathcal{P}_{\mathbb{T}^\perp}(\mathcal{Y}) - \mathcal{P}_{\mathbb{T}}(\mathcal{Y}), \mathcal{X}-\mathcal{T}\right\rangle \\ =\ &\langle \mathcal{P}_{\mathbb{T}^\perp}(\mathcal{M}), \mathcal{X}-\mathcal{T}\rangle - \left\langle \mathcal{P}_{\mathbb{T}}(\mathcal{Y}) - \mathcal{U} * \mathcal{V}^\top, \mathcal{X}-\mathcal{T}\right\rangle - \langle \mathcal{P}_{\mathbb{T}^\perp}(\mathcal{Y}), \mathcal{X}-\mathcal{T}\rangle \\ =\ &\|\mathcal{P}_{\mathbb{T}^\perp}(\mathcal{X}-\mathcal{T})\|_{\mathrm{TNN}} - \left\langle \mathcal{P}_{\mathbb{T}}(\mathcal{Y}) - \mathcal{U} * \mathcal{V}^\top, \mathcal{P}_{\mathbb{T}}(\mathcal{X}-\mathcal{T})\right\rangle - \langle \mathcal{P}_{\mathbb{T}^\perp}(\mathcal{Y}), \mathcal{P}_{\mathbb{T}^\perp}(\mathcal{X}-\mathcal{T})\rangle \\ \geq\ &\|\mathcal{P}_{\mathbb{T}^\perp}(\mathcal{X}-\mathcal{T})\|_{\mathrm{TNN}} - \left\|\mathcal{P}_{\mathbb{T}}(\mathcal{Y}) - \mathcal{U} * \mathcal{V}^\top\right\|_{\mathrm{F}} \|\mathcal{P}_{\mathbb{T}}(\mathcal{X}-\mathcal{T})\|_{\mathrm{F}} \\ &- \|\mathcal{P}_{\mathbb{T}^\perp}(\mathcal{Y})\|\, \|\mathcal{P}_{\mathbb{T}^\perp}(\mathcal{X}-\mathcal{T})\|_{\mathrm{TNN}} \\ \geq\ &\frac{1}{2}\|\mathcal{P}_{\mathbb{T}^\perp}(\mathcal{X}-\mathcal{T})\|_{\mathrm{TNN}} - \frac{1}{4}\sqrt{\frac{p}{n_3}}\cdot\sqrt{\frac{2n_3}{p}}\|\mathcal{P}_{\mathbb{T}^\perp}(\mathcal{X}-\mathcal{T})\|_{\mathrm{TNN}} \qquad (2.37)\\ \geq\ &\frac{1}{8}\|\mathcal{P}_{\mathbb{T}^\perp}(\mathcal{X}-\mathcal{T})\|_{\mathrm{TNN}} .\end{aligned}$$

Inequality (2.37) results from Condition 1 and Condition 2 and Lemma 2.14. Next, to verify the completeness of the proof, it suffices to show that $\|\mathcal{P}_{\mathbb{T}^\perp}(\mathcal{X}-\mathcal{T})\|_{\mathrm{TNN}}$ is strictly positive. We show it by contradiction. Suppose $\|\mathcal{P}_{\mathbb{T}^\perp}(\mathcal{X}-\mathcal{T})\|_{\mathrm{TNN}} = 0$, then $\mathcal{P}_{\mathbb{T}}(\mathcal{X}-\mathcal{T}) = \mathcal{X}-\mathcal{T}$ and $\mathcal{P}_{\mathbb{T}}\mathcal{R}_\Omega\mathcal{P}_{\mathbb{T}}(\mathcal{X}-$

$\mathcal{T}) = 0$. Therefore, we have

$$\|\mathcal{X} - \mathcal{T}\| = \|(\mathcal{P}_{\mathbb{T}}\mathcal{R}_{\Omega}\mathcal{P}_{\mathbb{T}} - \mathcal{P}_{\mathbb{T}})(\mathcal{X} - \mathcal{T})\| \leq \|\mathcal{P}_{\mathbb{T}}\mathcal{R}_{\Omega}\mathcal{P}_{\mathbb{T}} - \mathcal{P}_{\mathbb{T}}\|\|\mathcal{X} - \mathcal{T}\|,$$

which contradicts with the assumption that $\|\mathcal{P}_{\mathbb{T}}\mathcal{R}_{\Omega}\mathcal{P}_{\mathbb{T}} - \mathcal{P}_{\mathbb{T}}\| \leq \frac{1}{2}$. Thus, $\mathcal{T}$ is the unique minimizer to the optimization problem (2.18). □

Next we present a detailed proof of Theorem 2.6, our main theoretical result, demonstrating that our model ensures tensor recovery in high-probability.

2.6.3 Proof of Theorem 2.6

In this section, we provide a detailed proof of our main theoretical result Theorem 2.6. The proof is based on our Two-Step Tensor Completion (TSTC) algorithm. For the ease of the reader, we state the TSTC algorithm in Algorithm 2.7. This algorithm focuses on subtensor completion before combining results with t-CUR.

Algorithm 2.7 Two-Step Tensor Completion (TSTC)

1: **Input**: $[\mathcal{T}]_{\Omega_{\mathcal{R}} \cup \Omega_C}$: observed data; $\Omega_{\mathcal{R}}, \Omega_C$: observation locations; I, J: lateral and horizontal indices; r: target rank; TC: the chosen tensor completion solver.
2: $\widetilde{\mathcal{R}} = \text{TC}([\mathcal{T}]_{\Omega_{\mathcal{R}}}, r)$
3: $\widetilde{C} = \text{TC}([C]_{\Omega_C}, r)$
4: $\widetilde{\mathcal{U}} = [\widetilde{C}]_{I,:,:}$
5: $\widetilde{\mathcal{T}} = \widetilde{C} * \widetilde{\mathcal{U}}^{\dagger} * \widetilde{\mathcal{R}}$
6: **Output:** $\widetilde{\mathcal{T}}$: approximation of $\mathcal{T}$

Based on the idea of TSTC, it is crucial to understand that how the tensor incoherence properties of the original low tubal-rank tensor transfer to subtensors.

2.6.3.1 Incoherence passes to subtensors

Inspired by [17, Theorem 3.5], we explore how subtensors inherit the tensor incoherence conditions from the original tensor, differing from [99] in tensor norm and the definition of the tensor incoherence condition. Our focus is on subtensor incoherence due to its impact on the required sampling rate for accurate low tubal-rank tensor recovery (Theorem 2.5) and our emphasis on completing subtensors in tensor completion. We begin by examining the relationship between the tensor incoherence properties of subtensors and the original low tubal-rank tensor.

Lemma 2.15. *Let $\mathcal{T} \in \mathbb{K}^{n_1 \times n_2 \times n_3}$ satisfy the tensor μ_0-incoherence condition. Suppose that $\mathcal{T}$ has a compact t-SVD $\mathcal{T} = \mathcal{W} * \Sigma * \mathcal{V}^\top$ and a condition number κ. Consider the subtensors $\mathcal{C} = [\mathcal{T}]_{:,J,:}$ and $\mathcal{R} = [\mathcal{T}]_{I,:,:}$, each maintaining the same tubal-rank as $\mathcal{T}$. Their compact t-SVDs are represented as*

$$\mathcal{C} = \mathcal{W}_\mathcal{C} * \Sigma_\mathcal{C} * \mathcal{V}_\mathcal{C}^\top \text{ and } \mathcal{R} = \mathcal{W}_\mathcal{R} * \Sigma_\mathcal{R} * \mathcal{V}_\mathcal{R}^\top,$$

then the following results hold:

$$\mu_\mathcal{C} \le \kappa^2 \left\| [\mathcal{V}]_{J,:,:}^\dagger \right\|^2 \frac{|J|}{n_2} \mu_0, \text{ and } \mu_\mathcal{R} \le \kappa^2 \left\| [\mathcal{W}]_{I,:,:}^\dagger \right\|^2 \frac{|I|}{n_1} \mu_0.$$

Proof. First, let's prove that

$$\max_i \left\| [\widehat{\mathcal{W}_\mathcal{C}}^\top]_{:,:,k} \cdot \mathbf{e}_i \right\|_\mathrm{F} \le \sqrt{\frac{\mu_0 r}{n_1}}.$$

Notice that $\mathcal{C} = [\mathcal{T}]_{:,J,:} = \mathcal{W} * \Sigma * [\mathcal{V}]_{J,:,:}^\top$. Assume the compact t-SVD of $\Sigma * \left([\mathcal{V}]_{J,:,:}\right)^\top$ is $\Sigma * \left([\mathcal{V}]_{J,:,:}\right)^\top = \mathcal{P} * \mathcal{S} * \mathcal{Q}^\top$. Thus, $\mathcal{P} \in \mathbb{K}^{r \times r \times n_3}$ is an orthonormal tensor, leading to $\mathcal{W}_\mathcal{C} = \mathcal{W} * \mathcal{P}$ based on the relationship that $\mathcal{C} = \mathcal{W} * \mathcal{P} * \mathcal{S} * \mathcal{Q}^\top$.

$\mathcal{P}^\top * \mathcal{P} = \mathcal{I}$ implies that $[\widehat{\mathcal{P}}]_{:,:,k}^\top \cdot [\widehat{\mathcal{P}}]_{:,:,k} = \mathbb{I}_r$, where $\mathbb{I}_r$ is the $r \times r$ identity matrix for all $k \in [n_3]$. Therefore, we can establish that for $k \in [n_3]$,

$$\max_i \|\widehat{\mathcal{W}_\mathcal{C}}]_{:,:,k}^\top \cdot \mathbf{e}_i\|_\mathrm{F} = \max_i \|[\widehat{\mathcal{W}}]_{:,:,k}^\top \cdot \mathbf{e}_i\|_\mathrm{F} \le \sqrt{\frac{\mu_0 r}{n_1}}. \tag{2.38}$$

Next, let's prove $\max_i \|[\widehat{\mathcal{V}_\mathcal{C}}^\top]_{:,:,k} \cdot \mathbf{e}_i\|_\mathrm{F} \le \kappa(\mathcal{T}) \left\|([\mathcal{V}]_{J,:,:})^\dagger\right\| \sqrt{\frac{\mu_0 r}{n_2}}$. The compact t-SVD of $\mathcal{C}$ implies $\mathcal{V}_\mathcal{C}^\top = \Sigma_\mathcal{C}^\dagger * \mathcal{W}_\mathcal{C}^\top * \mathcal{C}$. Thus, for each $k \in [n_3]$, $[\widehat{\mathcal{V}_\mathcal{C}^\top}]_{:,:,k} = [\widehat{\Sigma}_\mathcal{C}^\dagger]_{:,:,k} \cdot [\widehat{\mathcal{W}_\mathcal{C}^\top}]_{:,:,k} \cdot [\widehat{\mathcal{C}}]_{:,:,k}$ holds and $\|[\widehat{\mathcal{V}_\mathcal{C}^\top}]_{:,:,k} \cdot \mathbf{e}_i\|_\mathrm{F}$ can be bounded by

$$\begin{aligned}
\|[\widehat{\mathcal{V}_\mathcal{C}^\top}]_{:,:,k} \cdot \mathbf{e}_i\|_\mathrm{F} =& \|[\widehat{\Sigma}_\mathcal{C}^\dagger]_{:,:,k} \cdot [\widehat{\mathcal{W}_\mathcal{C}^\top}]_{:,:,k} \cdot [\widehat{\mathcal{C}}]_{:,:,k} \cdot \mathbf{e}_i\|_\mathrm{F} \\
\le& \|[\widehat{\Sigma}_\mathcal{C}^\dagger]_{:,:,k}\| \|[\widehat{\mathcal{W}_\mathcal{C}}^\top]_{:,:,k} \cdot [\widehat{\mathcal{W}}]_{:,:,k} \cdot [\widehat{\Sigma}]_{:,:,k} \cdot [\widehat{\mathcal{V}}]_{J,:,k}^\top \cdot \mathbf{e}_i\|_\mathrm{F} \\
\le& \left\|\overline{\Sigma}_\mathcal{C}^\dagger\right\| \left\|\overline{\Sigma}\right\| \|[\widehat{\mathcal{V}}]_{J,:,k}^\top \cdot \mathbf{e}_i\|_\mathrm{F} \\
\le& \|\mathcal{C}^\dagger\| \, \|\mathcal{T}\| \, \|[\widehat{\mathcal{V}}]_{:,:,k}^\top \cdot \mathbf{e}_i\|_\mathrm{F} \\
=& \left\|([\mathcal{V}]_{J,:,:}^\top)^\dagger * \Sigma^\dagger * \mathcal{W}^\top\right\| \|\mathcal{T}\| \, \|[\widehat{\mathcal{V}}]_{:,:,k}^\top \cdot \mathbf{e}_i\|_\mathrm{F} \\
\le& \left\|([\mathcal{V}]_{J,:,:}^\top)^\dagger\right\| \|\mathcal{T}^\dagger\| \, \|\mathcal{T}\| \, \|[\widehat{\mathcal{V}}]_{:,:,k}^\top \cdot \mathbf{e}_i\|_\mathrm{F} \le \kappa \left\|([\mathcal{V}]_{J,:,:}^\top)^\dagger\right\| \sqrt{\frac{\mu_0 r}{n_2}}.
\end{aligned}$$

That is,

$$\max_i \|[\widehat{\mathcal{V}}_C^\top]_{:,:,k} \cdot \mathbf{e}_i\|_{\mathrm{F}} \le \kappa \left\|[\mathcal{V}]^\dagger_{J,:,:}\right\| \sqrt{\frac{\mu_0|J|}{n_2}} \sqrt{\frac{r}{|J|}}. \tag{2.39}$$

Combining (2.38) and (2.39), we can conclude that $\mu_C \le \kappa^2 \left\|[\mathcal{V}]^\dagger_{J,:,:}\right\|^2 \frac{|J|}{n_2}\mu_0$.

Applying above process on $\mathcal{R}$, we can get $\mu_{\mathcal{R}} \le \kappa^2 \left\|[\mathcal{W}]^\dagger_{I,:,:}\right\|^2 \frac{|I|}{n_1}\mu_0$. □

Following Lemma 2.15, we explore the incoherence properties of uniformly sampled subtensors, with summarized results below.

Lemma 2.16. *Let $\mathcal{T} \in \mathbb{K}^{n_1\times n_2\times n_3}$ with multi-rank $\vec{r}$, and let $\mathcal{T} = \mathcal{W} * \Sigma * \mathcal{V}^\top$ be its compact t-SVD. Additionally, $\mathcal{T}$ satisfies the tensor μ_0-incoherence condition, and κ denotes the condition number of $\mathcal{T}$. Suppose $I \subseteq [n_1]$ and $J \subseteq [n_2]$ are chosen uniformly at random with replacement. Then*

$$\mathrm{rank}_m(\mathcal{R}) = \mathrm{rank}_m(C) = \mathrm{rank}_m(\mathcal{T}), \mu_C \le \frac{25}{4}\kappa^2\mu_0 \text{ and } \mu_{\mathcal{R}} \le \frac{25}{4}\kappa^2\mu_0$$

hold with probability at least $1 - \frac{1}{n_1^\beta} - \frac{1}{n_2^\beta}$ provided that $|I| \ge 2\beta\mu_0\|\vec{r}\|_\infty \log(n_1\|\vec{r}\|_1)$ and $|J| \ge 2\beta\mu_0\|\vec{r}\|_\infty \log(n_2\|\vec{r}\|_1)$.

Proof of Lemma 2.16. According to Lemma 2.3 by setting $\delta = 0.815$ and $\beta_1 = \beta_2 = \beta$, we can easily get that

$$\mathbb{P}\left(\|[\mathcal{V}]^\dagger_{J,:,:}\| \le \sqrt{\frac{25n_2}{4|J|}}, \mathrm{rank}_m(C) = \mathrm{rank}_m(\mathcal{T})\right) \ge 1 - \frac{1}{n_2^\beta},$$

$$\mathbb{P}\left(\|[\mathcal{W}]^\dagger_{I,:,:}\| \le \sqrt{\frac{25n_1}{4|I|}}, \mathrm{rank}_m(\mathcal{R}) = \mathrm{rank}_m(\mathcal{T})\right) \ge 1 - \frac{1}{n_1^\beta}.$$

Therefore,

$$\mathbb{P}\left(\mu_C \le \frac{25}{4}\kappa^2\mu_0, \mathrm{rank}_m(C) = \mathrm{rank}_m(\mathcal{T})\right) \ge 1 - \frac{1}{n_2^\beta}, \tag{2.40}$$

$$\mathbb{P}\left(\mu_{\mathcal{R}} \le \frac{25}{4}\kappa^2\mu_0, \mathrm{rank}_m(\mathcal{R}) = \mathrm{rank}_m(\mathcal{T})\right) \ge 1 - \frac{1}{n_1^\beta}. \tag{2.41}$$

Combining (2.40) and (2.41), we can conclude that

$$\mathrm{rank}_m(\mathcal{R}) = \mathrm{rank}_m(C) = \mathrm{rank}_m(\mathcal{T}), \mu_C \le \frac{25}{4}\kappa^2\mu_0 \text{ and } \mu_{\mathcal{R}} \le \frac{25}{4}\kappa^2\mu_0$$

with probability at least $1 - \frac{1}{n_1^\beta} - \frac{1}{n_2^\beta}$ provided that

$$|I| \ge 2\beta\mu_0\|\vec{r}\|_\infty \log(n_1\|\vec{r}\|_1) \text{ and } |J| \ge 2\beta\mu_0\|\vec{r}\|_\infty \log(n_2\|\vec{r}\|_1).$$

□

2.6.3.2 Proof of Theorem 2.6

Proof of Theorem 2.6. Note that $I \subseteq [n_1]$ and $J \subseteq [n_2]$ are chosen uniformly with replacement. According to Lemma 2.16, we thus have

$$\operatorname{rank}_m(\mathcal{R}) = \operatorname{rank}_m(C) = \operatorname{rank}_m(\mathcal{T}), \mu_C \leq \frac{25}{4}\kappa^2\mu_0 \text{ and } \mu_{\mathcal{R}} \leq \frac{25}{4}\kappa^2\mu_0$$

hold with probability at least

$$\begin{aligned}&1 - \frac{1}{n_1^{800\beta\kappa^2\log(n_1n_3+n_2n_3)}} - \frac{1}{n_2^{800\beta\kappa^2\log(n_1n_3+n_2n_3)}}\\ =&1 - \frac{1}{(n_1n_3+n_2n_3)^{800\beta\kappa^2\log(n_1)}} - \frac{1}{(n_1n_3+n_2n_3)^{800\beta\kappa^2\log(n_2)}}\end{aligned}$$

provided that

$$|I| \geq 3200\beta\mu_0 r\kappa^2\log^2(n_1n_3+n_2n_3) \geq 800\kappa^2\log(n_1n_3+n_2n_3)\beta\cdot(2\mu_0\|\vec{r}\|_\infty\log(n_1\|\vec{r}\|_1)),$$

$$|J| \geq 3200\beta\mu_0 r\kappa^2\log^2(n_1n_3+n_2n_3) \geq 800\kappa^2\log(n_1n_3+n_2n_3)\beta\cdot(2\mu_0\|\vec{r}\|_\infty\log(n_2\|\vec{r}\|_1)).$$

Additionally, the following statements hold by Theorem 2.5 and the condition that $\mu_C \leq \frac{25}{4}\kappa^2\mu_0$ and $\mu_{\mathcal{R}} \leq \frac{25}{4}\kappa^2\mu_0$:

i) Given $C \in \mathbb{K}^{n_1\times|J|\times n_3}$ with $\operatorname{rank}(C) = r$,

$$p_C \geq \frac{1600\beta(n_1+|J|)\mu_0 r\kappa^2\log^2(n_1n_3+|J|n_3)}{n_1|J|}$$

for some $\beta > 1$ ensures that C is the unique minimizer to

$$\min_{\mathcal{X}\in\mathbb{K}^{n_1\times|J|\times n_3}} \|\mathcal{X}\|_{\text{TNN}}, \quad \text{subject to } \mathcal{P}_{\Omega_C}(\mathcal{X}) = \mathcal{P}_{\Omega_C}(C).$$

with probability at least $1 - \frac{3\log(n_1n_3+|J|n_3)}{(n_1n_3+|J|n_3)^{4\beta-2}}$.

ii) Given $\mathcal{R} \in \mathbb{K}^{|I|\times n_2\times n_3}$ with $\operatorname{rank}(\mathcal{R}) = r$,

$$p_{\mathcal{R}} \geq \frac{1600\beta(n_2+|I|)\mu_0 r\kappa^2\log^2(n_2n_3+|I|n_3)}{n_2|I|}$$

for some $\beta > 1$ ensures that $\mathcal{R}$ is the unique minimizer to

$$\min_{\mathcal{X}\in\mathbb{K}^{|I|\times n_2\times n_3}} \|\mathcal{X}\|_{\text{TNN}}, \quad \text{subject to } \mathcal{P}_{\Omega_{\mathcal{R}}}(\mathcal{X}) = \mathcal{P}_{\Omega_{\mathcal{R}}}(\mathcal{R}).$$

Once C and $\mathcal{R}$ are uniquely recovered from Ω_C and $\Omega_{\mathcal{R}}$, respectively. Then t-CUR decomposition can provide the reconstruction of $\mathcal{T}$ via $\mathcal{T} = C * \mathcal{U}^\dagger * \mathcal{R}$ with the condition $\operatorname{rank}_m(\mathcal{R}) = \operatorname{rank}_m(C) = \operatorname{rank}_m(\mathcal{T})$.

Combining all the statements above, we can conclude that $\mathcal{T}$ can be uniquely recovered from $\Omega_C \cup \Omega_R$ with probability at least

$$1 - \frac{2}{(n_1 n_3 + n_2 n_3)^{800\beta\kappa^2 \log(n_2)}} - \frac{3\log(n_1 n_3 + |J|n_3)}{(n_1 n_3 + |J|n_3)^{4\beta-2}} - \frac{3\log(n_2 n_3 + |I|n_3)}{(n_2 n_3 + |I|n_3)^{4\beta-2}}.$$

□

2.7 Conclusion

In this work, we present the t-CCS model, an extension of the matrix CCS model to a tensor framework. We provide both theoretical and experimental evidence demonstrating the flexibility and feasibility of the t-CCS model. The ITCURC algorithm, designed for the t-CCS model, provides a balanced trade-off between runtime efficiency and reconstruction quality. While it is not as effective as the state-of-the-art Bernoulli-Based TC algorithm, it is still comparable in terms of PSNR and SSIM. Thus, one of directions of our future research will focus on enhancing reconstruction quality through the integration of the M-product. From theoretical side, our current theoretical result shows that the t-CUR sampling scheme, as a special case of t-CCS model, requires a complexity of $O(\mu_0 r n_3(n_2 \log(n_1 n_3) + n_1 \log(n_2 n_3)))$ is sufficient, which is more sampling-efficient than that of a general t-CCS scheme. This finding suggests there is potential to further improve the theoretical sampling complexity for the t-CCS model, an aspect we plan to explore in our future work. Additionally, there is a need for comprehensive theoretical analysis on the convergence behavior of the ITCURC algorithm within the t-CCS model framework. Evaluating the algorithm's robustness against additive noise will also be a critical area of focus for future research. Furthermore, while our current work is limited to third-order tensors, we aim to extend our approach to accommodate higher-order tensor configurations in subsequent studies.

CHAPTER 3

ON THE ROBUSTNESS OF CROSS-CONCENTRATED SAMPLING FOR MATRIX COMPLETION

ABSTRACT

Matrix completion is essential in data science for recovering missing entries in partially observed data. Recently, cross-concentrated sampling (CCS), a novel approach to matrix completion, has gained attention, though its robustness against sparse outliers remains unaddressed. In this chapter, we propose the Robust CCS Completion problem to explore this robustness and introduce a non-convex iterative algorithm called Robust CUR Completion (RCURC). Our experiments with synthetic and real-world datasets confirm that RCURC is both efficient and robust to sparse outliers, making it a powerful tool for Robust Matrix Completion.

3.1 Introduction

The matrix completion problem, first introduced by Candes et al. [25] and Recht [97], aims to reconstruct a low-rank matrix X from a limited subset of its observed entries. In practice, many real-world data matrices are highly incomplete, and this problem has emerged as an important tool for uncovering latent structures in the data. The significance of matrix completion lies in its broad applicability across numerous domains such as recovering missing data in recommendation systems [7, 45], improving image quality [29, 61], and enhancing the efficiency and accuracy of signal processing techniques [22, 13].

In its simplest form, the goal of matrix completion is to estimate the unknown entries of a matrix X given access only to a small fraction of the entries. Mathematically, this can be described as solving for X given observations from the set Ω, which contains the indices of known entries in X. A common assumption is that the matrix X is of low rank, meaning that it can be described by a small number of underlying factors or components. The challenge arises from ensuring that the reconstructed matrix accurately captures the underlying low-rank structure without overfitting to the noise or sparsity of the observations.

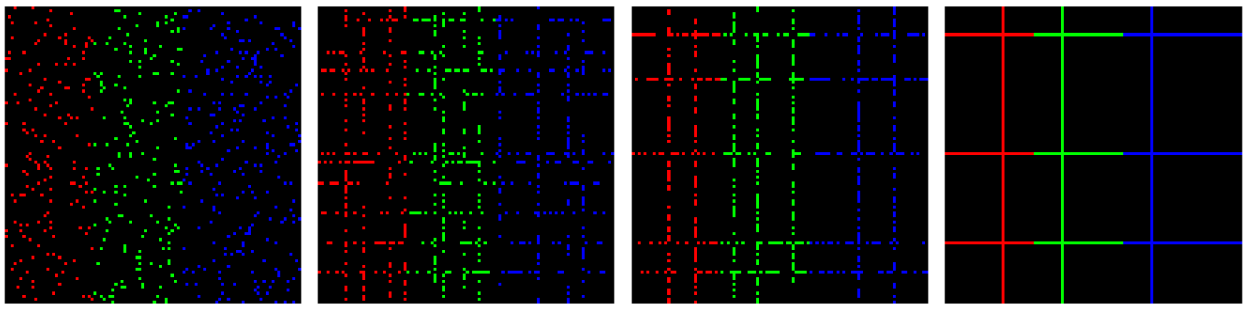
Traditional matrix completion methods often rely on uniform or Bernoulli sampling strategies, where each entry of the matrix is sampled independently with a fixed probability. However, this approach can be inefficient, particularly when the data exhibits specific structures or when some rows and columns are more informative than others.

Recent advancements in the matrix completion field have introduced a novel sampling method known as CCS [19]. Unlike uniform sampling, which treats all entries equally, CCS focuses on strategically sampling entries from certain rows and columns to achieve a more informative set of observations. By concentrating the samples in regions of the matrix that are more likely to contain useful information, CCS can lead to more accurate matrix recovery with fewer sampled entries.

Algorithm 3.1 outlines the key steps in the CCS procedure.

Algorithm 3.1 Cross-Concentrated Sampling (CCS) [19]

1: Input: the data matrix Y.
2: Uniformly select a subset of row indices I and column indices J.
3: Set $R = [Y]_{I,:}$ (rows indexed by I) and $C = [Y]_{:,J}$ (columns indexed by J).
4: Uniformly sample entries within the selected rows R and columns C, recording the sampled locations as Ω_{R} and Ω_{C}, respectively.
5: Output: Return the observed entries $[Y]_{\Omega_{\mathrm{R}} \cup \Omega_{\mathrm{C}}}$ and the indices Ω_{R}, Ω_{C}, I, J.



(a) Uniform Sampling (b) CCS–Less Concentrated (c) CCS–More Concentrated (d) CUR Sampling

Figure 3.1 [19] Visual comparison of sampling schemes: from uniform to CUR sampling at the same observation rate. Colored pixels indicate observed entries, black pixels indicate missing ones.

As shown in Figure 3.1, the CCS model bridges two commonly used sampling methods in matrix completion: Uniform Sampling and CUR Sampling. Uniform sampling randomly selects entries from the entire matrix, while CUR sampling focuses on selecting entire rows and columns for observation. The CCS approach can be viewed as a hybrid method, offering additional flexibility by concentrating samples on selected rows and columns, with a theoretical basis for achieving better recovery in certain structured datasets.

Despite the advantages of CCS, matrix completion in real-world applications often encounters a significant challenge: *data corruption by sparse outliers*. In many scenarios, the observed matrix is not simply low-rank but is also corrupted by noise or outliers that are sparsely distributed. Such outliers can arise from various sources, such as user input errors in recommendation systems or sensor malfunctions in signal processing. To address this, Robust Matrix Completion methods have been developed, which introduce a sparse matrix S to model the outliers, while ensuring that the underlying low-rank matrix X is accurately recovered.

A crucial question that remains is whether CCS-based matrix completion is robust to sparse outliers when used with robust recovery algorithms. Specifically, we ask:

Question 1. *[18]Is CCS-based matrix completion robust to sparse outliers under some robust algorithms, like the uniform sampling model?*

To address this, we examine the Robust CCS Completion problem, where we are given partial observations $\mathcal{P}_\Omega(Y)$ of a corrupted data matrix $Y = X + S$, where X is a low-rank matrix and S represents sparse outliers. The objective is to simultaneously recover both X and S using CCS-based sampling. The problem is formulated as follows:

$$\begin{aligned} \min_{X,S} \quad & \frac{1}{2}\langle \mathcal{P}_\Omega(X+S-Y), X+S-Y\rangle \\ \text{subject to} \quad & \operatorname{rank}(X) = r, \\ & S \text{ is } \alpha\text{-sparse}, \end{aligned}$$

where, the sampling operator $\mathcal{P}_\Omega$ is defined as:

$$\mathcal{P}_\Omega(Y) = \sum_{(i,j)\in\Omega} [Y]_{i,j} e_i e_j^\top,$$

where Ω denotes the set of observed indices generated by the CCS model. The sparse component S accounts for outliers, enabling more accurate recovery of the underlying low-rank matrix X. This framework extends the principles of Robust Matrix Completion, incorporating the novel CCS sampling approach.

3.1.1 Related Work

The problem of low-rank matrix recovery in the presence of sparse outliers has been well-studied under the settings of uniform sampling and Bernoulli sampling. This problem is known as *robust principal component analysis* (RPCA) when the corrupted data matrix is fully observed, and it is called Robust Matrix Completion if data is partially observed. The seminal work [23] considers both RPCA and Robust Matrix Completion problems via convex relaxed formulations and provides recovery guarantees. In particular, under the μ-incoherence assumption for the low-rank X, [23] requires the positions of outliers to be placed uniformly at random, and at least $0.1n^2$ entries are observed uniformly at random. Later, a series of non-convex algorithms [31, 78, 89, 132, 137, 12, 112, 20] tackle RPCA and/or Robust Matrix Completion problems with an improved, non-random α-sparsity assumptions for the outlier matrix S. The typical recovery guarantee shows a linear convergence of a non-convex algorithm, provided $\alpha \leq O(1/\text{poly}(\mu r))$; moreover, $O(\text{poly}(\mu r)\text{polylog}(n)n)$ random samples are typically required for the Robust Matrix Completion cases. Another line of work [30, 22, 136, 11, 13] focuses on the robust recovery of structured low-rank matrices, e.g., Hankel matrices, and they typically require merely $O(\text{poly}(\mu r)\text{polylog}(n))$ samples by utilizing the structure, even in the presence of structured outliers. More recently, [15, 17, 59] study the robust CUR decomposition problem, that is, recovering the low-rank matrix from row- and column-wise observations with entry-wise corruptions.

On the other hand, [19] shows that CCS-based matrix completion requires $O(\mu^2 r^2 n \log^2 n)$ samples which is only a factor of $\log n$ worse than the state-of-the-art result; however, its outlier tolerance has not been studied.

3.1.2 Notation

For a matrix M, $[M]_{i,j}$, $[M]_{I,:}$, $[M]_{:,J}$, and $[M]_{I,J}$ denote its (i, j)-th entry, its row submatrix with row indices I, its column submatrix with column indices J, and its submatrix with row indices I and column indices J, respectively. $M^\dagger$ represents the Moore–Penrose inverse of M. We use $\langle \cdot, \cdot \rangle$ to denote the Frobenius inner product. The symbol $[n]$ denotes the set $\{1, 2, \cdots, n\}$ for all $n \in \mathbb{Z}^+$. Throughout this chapter, uniform sampling is referred to as uniform sampling with replacement.

3.1.3 μ-incoherence and α-sparsity

In Robust Matrix Completion, we often rely on certain structural assumptions about the matrix to be recovered. Two such pivotal assumptions are μ-incoherence [26] and α-sparsity [23]. These assumptions play crucial roles in ensuring that the recovery algorithm can effectively reconstruct the matrix even when a significant portion of its entries are missing or corrupted by noise.

The concept of μ-incoherence is pivotal in the field of matrix completion, designed to ensure a balanced distribution of information across all rows and columns of a matrix. This balance is crucial for preventing any single row or column from disproportionately influencing the overall content of the matrix, which is particularly important when attempting to recover or approximate a matrix from a partial set of its entries.

Informally, a matrix is described as μ-incoherent when its singular vectors are such that no individual component dominates. This is quantified through boundedness conditions on the entries of the singular vectors, which ensure that the matrix's structural information is uniformly distributed.

We formalize this intuitive concept with the following definition:

Definition 21 (μ-incoherence [26])**.** *Let $X \in \mathbb{R}^{n_1 \times n_2}$ be a matrix of rank r. The matrix X is said to be μ-incoherent if the following conditions hold for its compact singular value decomposition $X = U\Sigma V^\top$:*

$$\|U\|_{2,\infty} \leq \sqrt{\frac{\mu r}{n_1}} \qquad \textit{and} \qquad \|V\|_{2,\infty} \leq \sqrt{\frac{\mu r}{n_2}},$$

where $\|\cdot\|_{2,\infty}$ represents the maximum ℓ_2-norm among the rows of the matrices U and V, respectively, and μ is a positive scalar that quantifies the level of incoherence.

This definition encapsulates how μ-incoherence functions as a safeguard against skewed data representation in matrix completion tasks, facilitating algorithms that require uniformly spread singular vectors for effective reconstruction.

The α-sparsity assumption plays a critical role in the analysis of matrix structures, particularly within the domain of Robust Matrix Completion. This assumption pertains to the density of non-zero entries in the matrix, or more specifically, in its constituent components such as the sparse error

matrix. The sparsity parameter α serves as a threshold, dictating the maximum allowable proportion of non-zero entries in each row and each column of the matrix, thus ensuring a controlled spread of these entries throughout the matrix.

Definition 22 (α-sparsity [23])**.** *Consider a matrix $S \in \mathbb{R}^{n_1 \times n_2}$. We define S as α-sparse if no more than an α fraction of its entries in each row and each column are non-zero. Specifically, for every row index i within the set of all row indices $[n_1]$ and every column index j within the set of all column indices $[n_2]$, the matrix satisfies:*

$$\|[S]_{i,:}\|_0 \leq \alpha n_2 \qquad \text{and} \qquad \|[S]_{:,j}\|_0 \leq \alpha n_1,$$

where $\|\cdot\|_0$ represent the number of non-zero entries.

The assumptions of μ-incoherence and α-sparsity are crucial because they directly influence the feasibility and complexity of the matrix recovery process in Robust Matrix Completion. These conditions ensure that the matrix has a well-distributed singular vector structure and a manageable number of outliers or corruptions, which are key for successful recovery using optimization-based methods [23, 26].

3.1.4 CUR Approximation

CUR approximation, also referred to as skeleton decomposition, forms the foundation of our algorithm design. To provide context, we will briefly review some key concepts of CUR approximation, which plays a crucial role in matrix dimensionality reduction.

CUR approximation is a powerful and interpretable technique that addresses the challenge of reducing matrix dimensionality while preserving meaningful structure. Given a rank-r matrix $X \in \mathbb{R}^{n \times n}$, the matrix can be reconstructed by selecting appropriate rows and columns that span its row and column spaces, respectively. This method offers an intuitive way to approximate matrices by extracting representative submatrices. The theoretical underpinnings of this approach have been established in prior research, as outlined in the following theorem:

Theorem 3.1 ([87, 58])**.** *Consider row and column index sets $I, J \subseteq [n]$. Denote the submatrices*

$C = [X]_{:,J}$, $U = [X]_{I,J}$, *and* $R = [X]_{I,:}$. *If* $\operatorname{rank}(U) = \operatorname{rank}(X)$, *then*

$$X = CU^{\dagger}R.$$

An extensive body of literature has significantly contributed to the development of sampling methods in matrix CUR approximation. Key works include those by Achlioptas and McSherry [2], Ahmadi and Drineas [3], Drineas et al. [41], Boutsidis and Woodruff [10], Hamm and Huang [58], Cai et al. [19], and Martinsson [88], among many others [76, 123, 35, 85, 126, 119, 87, 128, 111, 77, 67, 127, 36, 92, 106, 27, 44, 124, 93, 39, 134, 34, 46, 47, 118, 55, 75]. In fact, sampling a sufficient number of rows and columns makes this condition highly likely. An example of such a sampling strategy is provided in Theorem 3.2.

Theorem 3.2 ([32, Theorem 1.1])**.** *Let X satisfy Definition 21, and suppose we sample* $|I| = O(r \log n)$ *rows and* $|J| = O(r \log n)$ *columns uniformly at random. Then* $\operatorname{rank}(U) = \operatorname{rank}(X)$ *with probability at least* $1 - O(rn^{-2})$.

3.2 Proposed Algorithm

In this section, we introduce a novel non-convex algorithm for solving the Robust CCS Completion problem (3.1). The algorithm builds upon the projected gradient descent framework, integrating the CUR decomposition to efficiently compute low-rank approximations at each iteration. This approach significantly reduces computational complexity while maintaining robust performance. The proposed algorithm, named Robust CUR Completion (RCURC), is outlined in Algorithm 3.2.

RCURC leverages the CUR approximation, where selected rows and columns are used to capture the essential structure of the matrix. By iteratively updating both the sparse and low-rank components, RCURC aims to solve the matrix completion problem with high efficiency, even in the presence of sparse outliers. Specifically, in each iteration, the algorithm alternates between updating the sparse matrix using a thresholding operator and refining the low-rank matrix through projected gradient updates on the observed data. The method ensures that the low-rank component is efficiently approximated using the CUR decomposition, which focuses on key rows and columns to reduce the dimensionality of the problem.

Algorithm 3.2 Robust CUR Completion (RCURC)

Input: $[Y = X+S]_{\Omega_R \cup \Omega_C}$: observed data; Ω_R, Ω_C: observation locations; I, J: row and column indices that define R and C respectively; η_R, η_C: step sizes; r: target rank; ε: target precision level; ζ_0: initial thresholding value; γ: thresholding decay parameter.
$X_0 = M_0$; $\quad k = 0$
while $e_k > \varepsilon$ **do** // e_k is defined in (3.3)
  // Updating sparse component
  $\zeta_{k+1} = \gamma^k \zeta_0$
  $[S_{k+1}]_{I,:} = \mathcal{T}_{\zeta_{k+1}}[Y - X_k]_{I,:}$
  $[S_{k+1}]_{:,J} = \mathcal{T}_{\zeta_{k+1}}[Y - X_k]_{:,J}$
  // Updating low-rank component
  $R_{k+1} = [X_k]_{I,:} + \eta_R[\mathcal{P}_{\Omega_R}(Y - X_k - S_{k+1})]_{I,:}$
  $C_{k+1} = [X_k]_{:,J} + \eta_C[\mathcal{P}_{\Omega_C}(Y - X_k - S_{k+1})]_{:,J}$
  $U_{k+1} = \mathcal{H}_r([R_{k+1}]_{:,J} \uplus [C_{k+1}]_{I,:})$
  $[R_{k+1}]_{:,J} = U_{k+1}$ $\quad$ and $\quad$ $[C_{k+1}]_{I,:} = U_{k+1}$
  $X_{k+1} = C_{k+1}U_{k+1}^{\dagger}R_{k+1}$ // Do not compute (see (3.2))
  $k = k + 1$
end while
Output: C_k, U_k, R_k: The CUR components of the estimated low-rank matrix.

We will explain the algorithm step by step in the following paragraphs. For clarity, we begin with the low-rank component. To leverage the structure of cross-concentrated samples, it is efficient to enforce the low-rank constraint on X using the CUR approximation technique. Let $R = X_{I,:}$, $C = X_{:,J}$, and $U = X_{I,J}$. Applying gradient descent directly on R and C yields:

$$R_{k+1} = [X_k]_{I,:} + \eta_R \left[\mathcal{P}_{\Omega_R}(Y - X_k - S_{k+1})\right]_{I,:},$$

$$C_{k+1} = [X_k]_{:,J} + \eta_C \left[\mathcal{P}_{\Omega_C}(Y - X_k - S_{k+1})\right]_{:,J},$$

where η_R and η_C are the step sizes. However, When it comes to the intersection submatrix U, it is more complicated as Ω_R and Ω_C can have overlaps. We abuse the notation $\uplus$ and define an operator called *union sum* here:

$$\left[[R_{k+1}]_{:,J} \uplus [C_{k+1}]_{I,:}\right]_{i,j} = \begin{cases} [R_{k+1}]_{i,j} & \text{if } (i,j) \in \Omega_R \setminus \Omega_C; \\ [C_{k+1}]_{i,j} & \text{if } (i,j) \in \Omega_C \setminus \Omega_R; \\ \frac{\eta_R \eta_C}{\eta_R + \eta_C}\left(\frac{[R_{k+1}]_{i,j}}{\eta_R} + \frac{[C_{k+1}]_{i,j}}{\eta_C}\right) & \text{if } (i,j) \in \Omega_C \cap \Omega_R; \\ 0 & \text{otherwise.} \end{cases}$$

Basically, we take whatever value we have for the non-overlapped entries and take a weighted aver-

age for the overlaps where the weights are determined by the stepsizes used in the updates of R_{k+1} and C_{k+1}. To ensure the rank-r constraint, at least one of C_{k+1}, U_{k+1} or R_{k+1} should be rank-r. For computational efficiency, we choose to put it on the smallest one. Thus,

$$U_{k+1} = \mathcal{H}_r\left([R_{k+1}]_{:,J} \uplus [C_{k+1}]_{I,:}\right),$$

where $\mathcal{H}_r$ is the best rank-r approximation operator via truncated SVD. After replacing the intersection part U_{k+1} in the previously updated R_{k+1} and C_{k+1}, we have the new estimation of low-rank component:

$$X_{k+1} = C_{k+1}U_{k+1}^{\dagger}R_{k+1}. \tag{3.1}$$

However, (3.1) is just a conceptual step and one should never compute it. In fact, the full matrix X_k is never needed and should not be formed in the algorithm as updating the corresponding CUR components is sufficient.

We detect the outliers and put them into the sparse matrix S via hard-thresholding operator:

$$[\mathcal{T}_\zeta(M)]_{i,j} = \begin{cases} 0 & \text{if } |[M]_{i,j}| < \zeta; \\ [M]_{i,j} & \text{otherwise.} \end{cases}$$

The hard-thresholding on residue $Y - X_k$, paired with iterative decayed thresholding values:

$$\zeta_{k+1} = \gamma^k \zeta_0 \quad \text{with some } \gamma \in (0, 1),$$

has shown promising performance in outlier detection in prior art [12, 20, 14]. Notice that we only need to remove outliers located on the selected rows and columns, i.e., R and C, since they are the only components needed to update the low-rank component later. Therefore, for computational efficiency, we should only compute X_k on the selected rows and columns to update S_{k+1} correspondingly—as said, one should never compute the full X_k in this algorithm. In particular,

$$[X_k]_{I,:} = [C_k]_{I,:}U_k^{\dagger}R_k \;\text{ and }\; [X_k]_{:,J} = C_k U_k^{\dagger}[R_k]_{:,J}. \tag{3.2}$$

The stopping criteria is set to be $e_k \le \varepsilon$ where ε is the targeted accuracy and the computational error is

$$e_k = \frac{\langle \mathcal{P}_{\Omega_R \cup \Omega_C}(S_k + X_k - Y), S_k + X_k - Y\rangle}{\langle \mathcal{P}_{\Omega_R \cup \Omega_C} Y, Y\rangle}. \tag{3.3}$$

The recommended stepsizes are $\eta_R = \frac{1}{p_R}$ and $\eta_C = \frac{1}{p_C}$ where p_R and p_C are the observation rates of Ω_R and Ω_C respectively. Smaller stepsizes should be used with larger α, i.e., more outliers.

3.3 Numerical Experiments

In this section, we will verify the empirical performance of RCURC with both synthetic and real datasets. All experiments are implemented using MATLAB Online (R2024a, Update 6) and executed on a cloud-based Linux environment running Ubuntu 20.04 with kernel version 5.15.0-1062-AWS.

3.3.1 Synthetic Datasets

In this simulation, we assess the computational efficiency of our algorithm, RCURC, in addressing the robust CCS completion problem. We construct $Y = X + S$, a $d \times d$ matrix with $d = 3000$, where $X = WV^\top$ is a randomly generated rank-r matrix. To create the sparse outlier tensor S, we randomly select α percent entries to form the support of S. The values of the non-zero entries are then uniformly sampled from the range $[-c\mathbb{E}(\left|[X]_{i,j}\right|), c\mathbb{E}(\left|[X]_{i,j}\right|)]$. To generate the robust CCS completion problems, we set $\frac{|\mathrm{I}|}{d} = \frac{|\mathrm{J}|}{d} = 30\%$, and $\frac{|\Omega_R|}{|\mathrm{I}|d} = \frac{|\Omega_C|}{|\mathrm{J}|d} = 25\%$. The results are obtained by averaging over 10 runs and reported in Figure 3.2. Both figures in Figure 3.2 depict the relationship between the relative error e_k and computational time for our RCURC method with varying rank r and outlier amplification factor c. It is noteworthy that RCURC consistently achieves nearly linear convergence rates across different scenarios. The empirical convergence illustrated in the left subplot of Figure 3.2 shows that smaller r values allow the algorithm to achieve a given relative error in fewer iterations. This is likely because smaller r values minimize the impact of noise during the iterative process, enabling the algorithm to concentrate on the dominant low-rank structure of the matrix, which results in faster convergence.

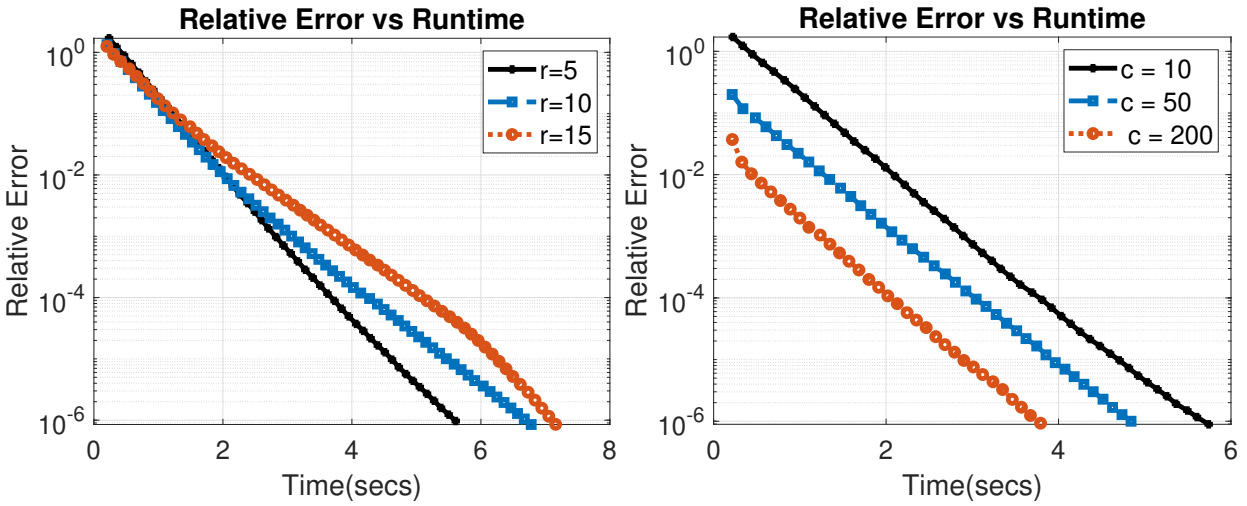


Figure 3.2 Empirical convergence of RCURC [18]. **Left:** c = 10 and varying r. **Right:** $\alpha = 0.2$, $r = 5$ and varying c.

3.3.2 Video Background Subtraction

We have applied RCURC under our robust CCS model to the problem of background separation. We evaluated our algorithm on the Train Station dataset [110]. The dataset is of size 173×216×500. In order to transform this data into a low-rank matrix, a specific reconfiguration process is applied. This process involves stacking each of the frontal slices of the tensor, which are essentially individual frames of the video. To transform high-dimensional video data into a low-rank matrix for streamlined data processing and analysis, we flatten the height and width dimensions (173 and 216) into a single dimension while retaining the frame dimension. This reshaping converts the original 3-dimensional tensor into a 2-dimensional matrix, facilitating subsequent computations. The resulting matrix has a size of 37,368 (the product of 173 and 216) by 500 (the number of frames). The CCS model is constructed by selecting 5% of the rows and 5% of the columns to create subrow and subcolumn matrices, with a sampling rate of 80% on each submatrices. We selected several benchmark algorithms for comparison, including Principal Component Pursuit (RPCA) [23], Stable Principal Component Pursuit (SPCP) [144], Low-Rank Matrix Factorization (LRMF) [81], Accelerated Alternating Projection (AccAltProj) [12], Iterated Robust CUR with fixed indices (IRCUR-F) [15], and Iterated Robust CUR with resampled indices (IRCUR-R) [15]. However, all six benchmark algorithms are operated under the full observation model since we find they do not visually perform

well on the constructed CCS mdoel. Since IRCUR-F and IRCUR-R are both CUR-based algorithms, we set the sampling parameters for IRCUR-R and IRCUR-F to 1.7732×10^2 and 4.0227, respectively. This configuration ensures that these two algorithms access 5% of the rows and 5% of the columns from the full observation data in each iteration. The methods RPCA, SPCP, AccAltProj, IRCUR-F, IRCUR-R and RCURC are configured to run **with a maximum of 50 iterations or until it meets a convergence tolerance of 10^{-3}, whichever condition is satisfied first.** We manually set the regularization parameter for RPCA and SPCP to 0.001, as it yielded the best visual results during tuning from the values 0.001, 0.01, 0.1, 1, and 10. The LRMF method is configured to run with a maximum of 5 iterations and a rank of 1. After manual tunings, where the rank parameter is tested with values 1, 3, and 5, we selected rank 1 as it provided the best balance between visual quality and runtime efficiency. Increasing the rank provided no significant improvement in visual quality while substantially increasing the runtime. Similarly, increasing the number of iterations beyond 5 does not result in noticeable improvements in visual quality, but significantly extended the runtime. For AccAltProj, IRCUR-R, IRCUR-F, and RCURC, we select a rank parameter of 1 as it yields the best visual results. This choice is based on tuning the rank parameter over the values $1, 3, 5, 7, 9$. The visual results are shown in Figure 3.3, consisting of five selected frames (80th, 160th, 240th, 320th and 400th). We present the corresponding quantitative results in Table 3.1, where the comparison is performed using the Peak Signal-to-Noise Ratio (PSNR) to evaluate reconstruction accuracy and computational time to assess efficiency. The PSNR is calculated by comparing reconstructed background, obtained through different methods, against the ground truth background tensor. The ground truth tensor is created by replicating the first frame, which represents a static background, across all frames in the dataset. For deterministic algorithms like RPCA, SPCP, LMRF, each quantitative result is calculated as the average of ten independent runs to help mitigate errors from machine precision, floating-point arithmetic, or other low-level numerical issues. In contrast, ICURC-R and ICURC-F involve randomness in row and column selection under full obeservation during iterations. Our method, based on the inherently random sampling model—the CCS model, naturally involves randomness. Thus, each quantitative result of

these three random method is computed as the average of ten independent runs, with the standard deviation included.

Table 3.1 Comparison of runtime and PSNR among RPCA, SPCP, LRMF, IRCUR-R, IRCUR-F based on full observation and RCURC based on the CCS model.

	RPCA	SPCP	LRMF	AccAltProj	IRCUR-R	IRCUR-F	RCURC
RUNTIME (SEC)	13.55	36.86	41.75	3.32	0.31 ± 0.02	0.18 ± 0.03	0.20 ± 0.05
PSNR	28.35	32.04	34.62	41.65	41.40 ± 0.19	41.06 ± 0.22	40.60 ± 0.58

The visualization results in Figure 3.3 illustrate the performance of various methods in reconstructing static background components. Rows 3, 4, and 5 highlight the effectiveness of RPCA, SPCP, and LRMF when applied to full observation. While these methods generally produce consistent and comparable results, subtle imperfections are noticeable in certain frames, such as the 160th frame, where minor blurring or incomplete restoration occurs. Rows 6, 7, and 8, corresponding to AccAltProj, IRCUR-R, and IRCUR-F, demonstrate notable performance in reconstructing background components. These methods effectively handle the background separation task, delivering visually satisfactory outputs. It is evident that the our method (last row) performs well in background subtraction. The results are comparable to other state-of-the-art algorithms under full observation, indicating the success of the our method in the video background separation task. The results in Table 3.1 provide a quantitative comparison of runtime and PSNR across various methods. The RCURC algorithm achieves a runtime of 0.20 ± 0.05 seconds, which is significantly faster than RPCA (13.55 seconds), SPCP (36.86 seconds), LRMF (41.75 seconds), and AccAltProj (3.32 seconds). When compared to IRCUR-R (0.31 ± 0.02 seconds) and IRCUR-F (0.18 ± 0.03 seconds), RCURC remains highly competitive in terms of runtime. Regarding PSNR, our method achieves a value of 40.60 ± 0.58, which is higher than RPCA (28.35), SPCP (32.04), and LRMF (34.62). Although the PSNR values of IRCUR-R (41.40 ± 0.19) and IRCUR-F (41.06 ± 0.22) are slightly higher, our method still demonstrates competitive performance.

Figure 3.3 **Video background subtraction results:** Row 1 shows the original images (full observation) at the corresponding frames, while Row 2 presents the observed images generated by the CCS model at the respective frames. Rows 3 to 8 showcase the background components extracted using RPCA, SPCP, LMRF, AccAltProj, IRCUR-R, and IRCUR-F algorithms based on the full observation model. Row 9 presents the results obtained using the RCURC algorithm under the CCS model.

3.4 Conclusion

This chapter introduces a novel mathematical model for robust matrix completion problems with cross-concentrated samples. A highly efficient non-convex algorithm, dubbed RCURC, has been developed for the proposed model. The key techniques are projected gradient descent and CUR approximation. The numerical experiments, with both synthetic and real datasets, show high potential. In particular, we consistently observe linear convergence on RCURC.

As for future work, we will study the statistical properties of the proposed robust CCS completion model, such as theoretical sample complexities and outlier tolerance. The recovery guarantee with a linear convergence rate will also be established for RCURC. We will also try to give a theoretical analysis explaining why a smaller rank accelerates the convergence of our RCURC algorithm. We will also explore other real-world applications that suit the proposed model.

CHAPTER 4

CONCLUSION

In this thesis, we have addressed critical challenges in matrix and tensor analysis by developing robust and flexible methodologies for data recovery tasks in data science. The methodologies propose in this thesis contribute to the advancement of matrix and tensor analysis, particularly in scenarios where robustness and flexibility are critical. By addressing the challenges posed by noise, sparsity, and complex data structures, the propose techniques have the potential to benefit a wide range of applications, such as image processing. Furthermore, the theoretical foundations established for robust sampling and decomposition provide a framework for future extensions in related fields. Our contributions span two interconnected projects, each tackling fundamental limitations in existing approaches while extending their applicability to real-world scenarios characterized by noise, sparsity, and high dimensionality. This chapter summarizes the key contributions of the thesis.

4.1 Summary of Contributions

Guaranteed Sampling Flexibility for Tensor Completion

In this project, we address the limitations of existing tensor completion methods by introducing Tensor Cross-Concentrated Sampling (t-CCS), a generalization of CCS to higher-order tensors. Accompanying this sampling framework, we develop the Iterative Tensor CUR Completion (ITCURC) algorithm, which offers theoretical guarantees for low-tubal-rank tensor recovery. Through rigorous theoretical analysis and extensive empirical validation, this project demonstrates the flexibility, accuracy, and computational efficiency of t-CCS-based model.

Robust CCS Completion for Matrix Analysis

In this project, we explore the robustness of CCS for matrix completion. While CCS has demonstrated effectiveness in capturing cross-sectional dependencies, its sensitivity to sparse outliers posed a significant limitation. To address this, we propose the Robust CCS Completion framework, introducing a non-convex iterative algorithm designed to handle noisy and incomplete data. Experiments on synthetic and real-world datasets validate our algorithm's efficiency and robustness, establishing it as a robust tool for practical data completion tasks.

BIBLIOGRAPHY

[1] Benefits and risks of MRI. Benefits and Risks of MRI. Accessed: 2023-12-19.

[2] D. Achlioptas and F. McSherry. Fast computation of low-rank matrix approximations. *Journal of the ACM (JACM)*, 54(2):9–es, 2007.

[3] S. Ahmadi-Asl, C. F. Caiafa, A. Cichocki, A. H. Phan, T. Tanaka, I. Oseledets, and J. Wang. Cross tensor approximation methods for compression and dimensionality reduction. *IEEE Access*, 9:150809–150838, 2021.

[4] S. Ahmadi-Asl, A. H. Phan, A. Cichocki, A. Sozykina, Z. A. Aghbari, J. Wang, and I. Oseledets. Adaptive cross tubal tensor approximation. *arXiv:2305.05030*, 2023.

[5] A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Machine learning*, 73(3):243–272, 2008.

[6] H. Avron and C. Boutsidis. Faster subset selection for matrices and applications. *SIAM Journal on Matrix Analysis and Applications.*, 34(4):1464–1499, 2013.

[7] J. Bennett, C. Elkan, B. Liu, P. Smyth, and D. Tikk. KDD cup and workshop 2007. *SIGKDD Explor. Newsl.*, 9(2):51–52, 2007.

[8] A. Bhaskara, A. Rostamizadeh, J. Altschuler, M. Zadimoghaddam, T. Fu, and V. Mirrokni. Greedy column subset selection: New bounds and distributed algorithms. International Conference on Machine Learning, 2016.

[9] C. Boutsidis, P. Drineas, and M. Magdon-Ismail. Near-optimal column-based matrix reconstruction. *SIAM Journal on Computing*, 43(2):687–717, 2014.

[10] C. Boutsidis and D. Woodruff. Optimal CUR matrix decompositions. *SIAM Journal on Computing*, 46(2):543–589, 2017.

[11] H. Cai, J.-F. Cai, T. Wang, and G. Yin. Accelerated structured alternating projections for robust spectrally sparse signal recovery. *IEEE Transactions on Signal Processing*, 69:809–821, 2021.

[12] H. Cai, J.-F. Cai, and K. Wei. Accelerated alternating projections for robust principal component analysis. *Journal of Machine Learning Research*, 20(1):685–717, 2019.

[13] H. Cai, J.-F. Cai, and J. You. Structured gradient descent for fast robust low-rank Hankel matrix completion. *SIAM Journal on Scientific Computing.*, 45(3):A1172–A1198, 2023.

[14] H. Cai, Z. Chao, L. Huang, and D. Needell. Fast robust tensor principal component analysis via fiber CUR decomposition. In *Proceedings of the IEEE/CVF International Conference*

on Computer Vision (ICCV) Workshops, pages 189–197, 2021.

[15] H. Cai, K. Hamm, L. Huang, J. Li, and T. Wang. Rapid robust principal component analysis: CUR accelerated in exact low rank estimation. *IEEE Signal Processing Letters*, 28:116–120, 2020.

[16] H. Cai, K. Hamm, L. Huang, and D. Needell. Mode-wise tensor decompositions: Multi-dimensional generalizations of CUR decompositions. *Journal of Machine Learning Research*, 22(185):1–36, 2021.

[17] H. Cai, K. Hamm, L. Huang, and D. Needell. Robust CUR decomposition: Theory and imaging applications. *SIAM Journal on Imaging Sciences*, 14(4):1472–1503, 2021.

[18] H. Cai, L. Huang, C. Kundu, and B. Su. On the robustness of cross-concentrated sampling for matrix completion. In *2024 58th Annual Conference on Information Sciences and Systems (CISS)*, pages 1–5, 2024.

[19] H. Cai, L. Huang, P. Li, and D. Needell. Matrix completion with cross-concentrated sampling: Bridging uniform sampling and CUR sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[20] H. Cai, J. Liu, and W. Yin. Learned robust PCA: A scalable deep unfolding approach for high-dimensional outlier detection. In *Advances in Neural Information Processing Systems*, volume 34, pages 16977–16989, 2021.

[21] J.-F. Cai, R. H. Chan, and Z. Shen. A framelet-based image inpainting algorithm. *Applied and Computational Harmonic Analysis*, 24(2):131–149, 2008.

[22] J.-F. Cai, T. Wang, and K. Wei. Fast and provable algorithms for spectrally sparse signal reconstruction via low-rank Hankel matrix completion. *Applied and Computational Harmonic Analysis*, 46(1):94–121, 2019.

[23] E. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM*, 58(3):1–37, 2011.

[24] E. Candes and B. Recht. Exact matrix completion via convex optimization. *Communications of the ACM*, 55(6):111–119, 2012.

[25] E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.

[26] E. J. Candès and T. Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE transactions on information theory*, 56(5):2053–2080, 2010.

[27] C. Chen, M. Gu, Z. Zhang, W. Zhang, and Y. Yu. Efficient spectrum-revealing cur matrix

decomposition. In *International Conference on Artificial Intelligence and Statistics*, pages 766–775. Proceedings of Machine Learning Research, 2020.

[28] J. Chen, Y. Wei, and Y. Xu. Tensor CUR decomposition under t-product and its perturbation. *Numerical Functional Analysis and Optimization*, 43(6):698–722, 2022.

[29] P. Chen and D. Suter. Recovering the missing components in a large noisy low-rank matrix: Application to sfm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(8):1051–1063, 2004.

[30] Y. Chen and Y. Chi. Robust spectral compressed sensing via structured matrix completion. *IEEE Transactions on Information Theory*, 60(10):6576–6601, 2014.

[31] Y. Chen, A. Jalali, S. Sanghavi, and C. Caramanis. Low-rank matrix recovery from errors and erasures. *IEEE Transactions on Information Theory*, 59(7):4324–4337, 2013.

[32] J. Chiu and L. Demanet. Sublinear randomized algorithms for skeleton decompositions. *SIAM Journal on Matrix Analysis and Applications.*, 34(3):1361–1383, 2013.

[33] A. Cichocki, N. Lee, I. Oseledets, A.-H. Phan, Q. Zhao, D. P. Mandic, et al. Tensor networks for dimensionality reduction and large-scale optimization: Part 1 low-rank tensor decompositions. *Foundations and Trends® in Machine Learning*, 9(4-5):249–429, 2016.

[34] K. L. Clarkson and D. Woodruff. Numerical linear algebra in the streaming model. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 205–214, 2009.

[35] A. Deshpande and L. Rademacher. Efficient volume sampling for row/column subset selection. In *2010 ieee 51st annual symposium on foundations of computer science*, pages 329–338. IEEE, 2010.

[36] Y. Dong and P.-G. Martinsson. Simpler is better: A comparative study of randomized algorithms for computing the CUR decomposition. *arXiv preprint arXiv:2104.05877*, 2021.

[37] D. L. Donoho. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006.

[38] P. Drineas, R. Kannan, and M. Mahoney. Fast Monte Carlo algorithms for matrices III: Computing a compressed approximate matrix decomposition. *SIAM Journal on Computing*, 36(1):184–206, 2006.

[39] P. Drineas, R. Kannan, and M. Mahoney. Fast monte carlo algorithms for matrices iii: Computing a compressed approximate matrix decomposition. *SIAM Journal on Computing*, 36(1):184–206, 2006.

[40] P. Drineas, M. Mahoney, and S. Muthukrishnan. Relative-error CUR matrix decompositions. *SIAM Journal on Matrix Analysis and Applications.*, 30(2):844–881, 2008.

[41] P. Drineas, M. Mahoney, and S. Muthukrishnan. Relative-error CUR matrix decompositions. *SIAM Journal on Matrix Analysis and Applications*, 30(2):844–881, 2008.

[42] G. Ely, S. Aeron, N. Hao, and M. Kilmer. 5D seismic data completion and denoising using a novel class of tensor decompositions. *Geophysics*, 80:V83 – V95, 2015.

[43] A. Gaur and S. S. Gaur. *Statistical methods for practice and research: A guide to data analysis using SPSS*. Sage, 2006.

[44] P. Y. Gidisu and M. E. Hochstenbach. A hybrid DEIM and leverage scores based method for CUR index selection. In *European Consortium for Mathematics in Industry*, pages 147–153. Springer, 2021.

[45] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, 1992.

[46] S. A. Goreinov, I. Oseledets, D. V. Savostyanov, E. E. Tyrtyshnikov, and N. L. Zamarashkin. How to find a good submatrix. In *Matrix Methods: Theory, Algorithms And Applications: Dedicated to the Memory of Gene Golub*, pages 247–256. World Scientific, 2010.

[47] S. A. Goreinov and E. E. Tyrtyshnikov. The maximal-volume concept in approximation by low-rank matrices. *Contemporary Mathematics*, 280:47–52, 2001.

[48] L. Grasedyck. Hierarchical singular value decomposition of tensors. *SIAM Journal on Matrix Analysis and Applications.*, 31(4):2029–2054, 2010.

[49] L. Grasedyck and S. Krämer. Stable ALS approximation in the TT-format for rank-adaptive tensor completion. *Numerische Mathematik*, 143(4):855–904, 2019.

[50] D. Gross. Recovering low-rank matrices from few coefficients in any basis. *IEEE Transactions on Information Theory*, 57:1548–1566, 2009.

[51] W. Guo and J.-M. Qiu. A local macroscopic conservative (lomac) low rank tensor method for the vlasov dynamics. *arXiv preprint arXiv:2207.00518*, 2022.

[52] W. Guo and J.-M. Qiu. A low rank tensor representation of linear transport and nonlinear vlasov solutions and their associated flow maps. *Journal of Computational Physics*, 458:111089, 2022.

[53] W. Guo and J.-M. Qiu. A conservative low rank tensor method for the Vlasov dynamics. 46(1):A232–A263, 2024.

[54] W. Hackbusch and S. Kühn. A new scheme for the tensor representation. *Journal of Fourier analysis and applications*, 15(5):706–722, 2009.

[55] N. Halko, P.-G. Martinsson, and J. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.

[56] K. Hamm. Generalized pseudoskeleton decompositions. *Linear Algebra and Its Applications*, 664:236–252, 2023.

[57] K. Hamm and L. Huang. Perspectives on CUR decompositions. *Applied and Computational Harmonic Analysis*, 48(3):1088–1099, 2020.

[58] K. Hamm and L. Huang. Stability of sampling for CUR decompositions. *Foundations of Data Science*, 2(2):83, 2020.

[59] K. Hamm, M. Meskini, and H. Cai. Riemannian CUR decompositions for robust principal component analysis. In *ICML Workshop on Topology, Algebra, and Geometry in Machine Learning*, 2022.

[60] F. Hitchcock. The expression of a tensor or a polyadic as a sum of products. *Journal of Mathematical Physics*, 6(1-4):164–189, 1927.

[61] Y. Hu, D. Zhang, J. Ye, X. Li, and X. He. Fast and accurate matrix completion via truncated nuclear norm regularization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(9):2117–2130, 2012.

[62] P. Jain and S. Oh. Provable tensor factorization with missing data. In *Advances in Neural Information Processing Systems*, volume 2, page 1431 – 1439, 2014.

[63] S. Jain, A. Gutierrez, and J. Haupt. Noisy tensor completion for tensors with a sparse canonical polyadic factor. In *IEEE International Symposium on Information Theory*, pages 2153–2157, 2017.

[64] T. Jiang, T. Huang, X. Zhao, and L. Deng. Multi-dimensional imaging data recovery via minimizing the partial sum of tubal nuclear norm. *Journal of Computational and Applied Mathematics*, 372:112680, 2020.

[65] T. Jiang, T. Huang, X. Zhao, T. Ji, and L. Deng. Matrix factorization for low-rank tensor completion using framelet prior. *Information Sciences*, 436:403–417, 2018.

[66] T. Jiang, M. K. P. Ng, X. Zhao, and T. Huang. Framelet representation of tensor nuclear norm for third-order tensor completion. *IEEE Transactions on Image Processing*, 29:7233–7244, 2020.

[67] R. Jin and S. Zhu. CUR algorithm with incomplete matrix observation. *arXiv preprint arXiv:1403.5647*, 2014.

[68] H. Johan. Tensor rank is NP-complete. *Journal of Algorithms*, 4(11):644–654, 1990.

[69] M. Kilmer, K. Braman, N. Hao, and R. C. Hoover. Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging. *SIAM Journal on Matrix Analysis and Applications.*, 34(1):148–172, 2013.

[70] M. Kilmer, L. Horesh, H. Avron, and E. Newman. Tensor-tensor products for optimal representation and compression. *arXiv:2001.00046*, 2019.

[71] M. Kilmer and C. Martin. Factorization strategies for third-order tensors. *Linear Algebra and Its Applications*, 435(3):641–658, 2011.

[72] A. Kolbeinsson, J. Kossaifi, Y. Panagakis, A. Bulat, A. Anandkumar, I. Tzoulaki, and P. M. Matthews. Tensor dropout for robust learning. *IEEE Journal of Selected Topics in Signal Processing*, 15(3):630–640, 2021.

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

[74] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[75] S. Kumar, M. Mohri, and A. Talwalkar. Ensemble nystrom method. *Advances in Neural Information Processing Systems*, 22, 2009.

[76] S. Kumar, M. Mohri, and A. Talwalkar. Sampling methods for the Nyström method. *Journal of Machine Learning Research*, 13(1):981–1006, 2012.

[77] C. Li, X. Wang, W. Dong, J. Yan, Q. Liu, and H. Zha. Joint active learning with feature selection via CUR matrix decomposition. *IEEE transactions on pattern analysis and machine intelligence*, 41(6):1382–1396, 2018.

[78] X. Li. Compressed sensing and matrix completion with constant proportion of corruptions. *Constructive Approximation*, 37:73–99, 2013.

[79] X. Li and Y. Pang. Deterministic column-based matrix decomposition. *IEEE Transactions on Knowledge and Data Engineering*, 22(1):145–149, 2010.

[80] J. Liu, P. Musialski, P. Wonka, and J. Ye. Tensor completion for estimating missing values in visual data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):208–220, 2013.

[81] Q. Liu and X. Li. Efficient low-rank matrix factorization based on $\ell_{1,\varepsilon}$-norm for online background subtraction. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(7):4900–4904, 2021.

[82] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan. Tensor robust principal component analysis: Exact recovery of corrupted low-rank tensors via convex optimization. In *Proceedings of the IEEE conference on computer vision and pattern Recognition*, pages 5249–5257, 2016.

[83] 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 Transactions on Pattern Analysis and Machine Intelligence*, 42(04):925–938, apr 2020.

[84] C. Lu, J. Feng, Z. Lin, and S. Yan. Exact low tubal rank tensor recovery from Gaussian measurements. In *Proc. 27th International Joint Conference on Artificial Intelligence*, pages 2504–2510, 2018.

[85] L. Mackey, M. Jordan, and A. Talwalkar. Divide-and-conquer matrix factorization. *Advances in neural information processing systems*, 24, 2011.

[86] M. Mahoney and P. Drineas. CUR matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences of the United States of America*, 106(3):697–702, 2009.

[87] M. Mahoney and P. Drineas. CUR matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences of the United States of America*, 106(3):697–702, 2009.

[88] P.-G. Martinsson and J. Tropp. Randomized numerical linear algebra: Foundations and algorithms. *Acta Numerica*, 29:403–572, 2020.

[89] P. Netrapalli, U. Niranjan, S. Sanghavi, A. Anandkumar, and P. Jain. Non-convex robust PCA. In *Advances in Neural Information Processing Systems*, pages 1107–1115, 2014.

[90] L. Omberg, G. H. Golub, and O. Alter. A tensor higher-order singular value decomposition for integrative analysis of DNA microarray data from different studies. *Proceedings of the National Academy of Sciences of the United States of America*, 104(47):18371–18376, 2007.

[91] I. Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.

[92] U. Oswal, S. Jain, K. S. Xu, and B. Eriksson. Block CUR: Decomposing matrices using groups of columns. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2018, Dublin, Ireland, September 10–14, 2018, Proceedings, Part II 18*, pages 360–376. Springer, 2019.

[93] V. Y. Pan, Q. Luan, J. Svadlenka, and L. Zhao. Superfast CUR matrix algorithms, their pre-processing and extensions. *arXiv preprint arXiv:1710.07946*, 2017.

[94] R. Peng and E. Matsui. *The Art of Data Science: A guide for anyone who works with Data*. Skybrude Consulting, LLC, 2015.

[95] J. Popa, S. Minkoff, and Y. Lou. An improved seismic data completion algorithm using low-rank tensor optimization: Cost reduction and optimal data orientation. *Geophysics*, 86(3):V219–V232, 2021.

[96] W. Qin, H. Wang, F. Zhang, J. Wang, X. Luo, and T. Huang. Low-rank high-order tensor completion with applications in visual data. *IEEE Transactions on Image Processing*, 31:2433–2448, 2022.

[97] B. Recht. A simpler approach to matrix completion. *Journal of Machine Learning Research*, 12(12), 2011.

[98] S. Rendle. Factorization machines. In *2010 IEEE International conference on data mining*, pages 995–1000. IEEE, 2010.

[99] M. M. Salut and D. Anderson. Tensor robust CUR for compression and denoising of hyperspectral data. *IEEE Access*, 2023.

[100] M. M. Salut and D. V. Anderson. Tensor robust cur for compression and denoising of hyperspectral data. *IEEE Access*, 2023.

[101] P. Shah, N. Rao, and G. Tang. Sparse and low-rank tensor decomposition. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

[102] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos. Tensor decomposition for signal processing and machine learning. *IEEE Transactions on signal processing*, 65(13):3551–3582, 2017.

[103] A. Sobral and E. Zahzah. Matrix and tensor completion algorithms for background model initialization: A comparative evaluation. *Pattern Recognition Letters*, 96:22–33, 2017.

[104] G. Song, M. K. P. Ng, and X. Zhang. Robust tensor completion using transformed tensor singular value decomposition. *Numerical Linear Algebra with Applications*, 27(3):e2299, 2020.

[105] G. Song, M. K. P. Ng, and X. Zhang. Tensor completion by multi-rank via unitary transformation. *Applied and Computational Harmonic Analysis*, 65:348–373, 2023.

[106] D. C. Sorensen and M. Embree. A deim induced CUR factorization. *SIAM Journal on Scientific Computing*, 38(3):A1454–A1482, 2016.

[107] B. Su, J. You, H. Cai, and L. Huang. Guaranteed sampling flexibility for low-tubal-rank tensor completion. *arXiv preprint arXiv:2406.11092*, 2024.

[108] Z. Tan, L. Huang, H. Cai, and Y. Lou. Non-convex approaches for low-rank tensor completion under tubal sampling. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, 2023.

[109] D. A. Tarzanagh and G. Michailidis. Fast randomized algorithms for t-product based tensor operations and decompositions with applications to imaging data. *SIAM Journal on Imaging Sciences*, 11(4):2629–2664, 2018.

[110] D. Thirde, L. Li, and F. Ferryman. Overview of the PETS2006 challenge. In *Proc. 9th IEEE international workshop on performance evaluation of tracking and surveillance (PETS 2006)*, pages 47–50, 2006.

[111] C. Thurau, K. Kersting, and C. Bauckhage. Deterministic CUR for improved large-scale data analysis: An empirical study. In *Proceedings of the 2012 SIAM International Conference on Data Mining*, pages 684–695. SIAM, 2012.

[112] T. Tong, C. Ma, and Y. Chi. Accelerating ill-conditioned low-rank matrix estimation via scaled gradient descent. *Journal of Machine Learning Research*, 22(150):1–63, 2021.

[113] T. Tong, C. Ma, A. Prater-Bennette, E. Tripp, and Y. Chi. Scaling and scalability: Provable nonconvex low-rank tensor estimation from incomplete measurements. *Journal of Machine Learning Research*, 23(163):1–77, 2022.

[114] J. Tropp. Column subset selection, matrix factorization, and eigenvalue optimization. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 978–986. Society for Industrial and Applied Mathematics, 2009.

[115] J. Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12:389–434, 2010.

[116] J. Tropp. An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2):1–230, 2015.

[117] L. Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.

[118] E. Tyrtyshnikov. Incomplete cross approximation in the mosaic-skeleton method. *Computing*, 64:367–380, 2000.

[119] S. Voronin and P.-G. Martinsson. Efficient algorithms for CUR and interpolative matrix decompositions. *Advances in Computational Mathematics*, 43:495–516, 2017.

[120] A. Wang and Z. Jin. Near-optimal noisy low-tubal-rank tensor completion via singular tube thresholding. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 553–560, 2017.

[121] A. Wang, D. Wei, B. Wang, and Z. Jin. Noisy low-tubal-rank tensor completion through iterative singular tube thresholding. *IEEE Access*, 6:35112–35128, 2018.

[122] S. Wang and Z. Zhang. Improving CUR matrix decomposition and the nyström approximation via adaptive sampling. *Journal of Machine Learning Research*, 14(1):2729–2769, 2013.

[123] S. Wang and Z. Zhang. Improving CUR matrix decomposition and the nyström approximation via adaptive sampling. *Journal of Machine Learning Research*, 14(1):2729–2769, 2013.

[124] S. Wang, Z. Zhang, and T. Zhang. Towards more efficient spsd matrix approximation and CUR matrix decomposition. *Journal of Machine Learning Research*, 17(209):1–49, 2016.

[125] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[126] D. Woodruff et al. Sketching as a tool for numerical linear algebra. *Foundations and Trends® in Theoretical Computer Science*, 10(1–2):1–157, 2014.

[127] C. Wu, H. Zhao, and J. Hu. Near field sampling compression based on matrix CUR decomposition. In *2021 IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting (APS/URSI)*, pages 1455–1456. IEEE, 2021.

[128] M. Xu, R. Jin, and Z.-H. Zhou. CUR algorithm for partially observed matrices. In *International Conference on Machine Learning*, pages 1412–1421. Proceedings of Machine Learning Research, 2015.

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

[130] S. Xue, W. Qiu, F. Liu, and X. Jin. Low-rank tensor completion by truncated nuclear norm regularization. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 2600–2605. IEEE, 2018.

[131] J. Yang, X. Zhao, T. Ma, Y. Chen, T. Huang, and M. Ding. Remote sensing images destriping using unidirectional hybrid total variation and nonconvex low-rank regularization. *Journal of Computational and Applied Mathematics*, 363:124–144, 2020.

[132] X. Yi, D. Park, Y. Chen, and C. Caramanis. Fast algorithms for robust PCA via gradient

descent. In *Advances in Neural Information Processing Systems*, pages 4152–4160, 2016.

[133] F. Zhang, J. Wang, W. Wang, and C. Xu. Low-tubal-rank plus sparse tensor recovery with prior subspace information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(10):3492–3507, 2020.

[134] G. Zhang, H. Li, and Y. Wei. CPQR-based randomized algorithms for generalized cur decompositions. *Computational and Applied Mathematics*, 43(3):132, 2024.

[135] L. Zhang, L. Song, B. Du, and Y. Zhang. Nonlocal low-rank tensor completion for visual data. *IEEE Transactions on Cybernetics*, 51(2):673–685, 2019.

[136] S. Zhang and M. Wang. Correction of corrupted columns through fast robust hankel matrix completion. *IEEE Transactions on Signal Processing*, 67(10):2580–2594, 2019.

[137] T. Zhang and Y. Yang. Robust PCA by manifold optimization. *Journal of Machine Learning Research*, 19(1):3101–3139, 2018.

[138] Z. Zhang and S. Aeron. Exact tensor completion using t-SVD. *IEEE Transactions on Signal Processing*, 65(6):1511–1526, 2017.

[139] Z. Zhang, G. Ely, S. Aeron, N. Hao, and M. Kilmer. Novel methods for multilinear data completion and de-noising based on tensor-SVD. In *Proceedings of the IEEE conference on computer vision and pattern Recognition*, pages 3842–3849, 2014.

[140] Q. Zhao, L. Zhang, and A. Cichocki. Bayesian CP factorization of incomplete tensors with automatic rank determination. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1751–1763, 2015.

[141] X. Zhao, J. Yang, T. Ma, T. Jiang, M. K. P. Ng, and T. Huang. Tensor completion via complementary global, local, and nonlocal priors. *IEEE Transactions on Image Processing*, 31:984–999, 2022.

[142] Y. Zheng, T. Huang, X. Zhao, T. Jiang, T. Ma, and T. Ji. Mixed noise removal in hyperspectral image via low-fibered-rank regularization. *IEEE Transactions on Geoscience and Remote Sensing*, 58(1):734–749, 2019.

[143] P. Zhou, C. Lu, Z. Lin, and C. Zhang. Tensor factorization for low-rank tensor completion. *IEEE Transactions on Image Processing*, 27(3):1152–1163, 2017.

[144] Z. Zhou, X. Li, J. Wright, E. Candes, and Y. Ma. Stable principal component pursuit. In *2010 IEEE international symposium on information theory*, pages 1518–1522. IEEE, 2010.