

ON METHODS IN TENSOR RECOVERY AND COMPLETION

By

Cullen Haselby

A DISSERTATION

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

Applied Mathematics—Doctor of Philosophy

2023

## ABSTRACT

Tensor representations of data have great promise, since as the size of data grows both in terms of dimensionality and modes, it becomes increasingly advantageous to employ methods that exploit structure to explain, store, and compute in an efficient manner. Tensors and their decompositions crucially admit the possibility of using multi-linear relationships between the modes to exploit for such purposes. Additionally multi-modal views of data are natural in many practical contexts; e.g. physical phenomenon may have several spatial modes and a temporal mode, in natural language processing data may derive from triplets of words that form subject-verb-object semantics, in neural networks both processing data and storing weights between layers have been fruitfully imagined using tensors.

Finding or using decompositions of tensors is a fundament of this work, and indeed any work which seeks to analyze and produce algorithms which make efficient use of tensor representations of data since these factorizations can radically lower the total number of parameters required to store and perform operations on the tensor. In this dissertation we consider two related problems involving tensor representations of data.

First, we consider the problem of recovering a low-rank Tucker approximation to a large tensor based on structured, yet randomized, measurements. We will describe a framework for a class of measurement ensembles that can offer excellent memory and accuracy trade-offs in comparison to alternatives. Furthermore, these ensembles can easily be applied in a single pass over the tensor, and in a linear manner making them suitable for streaming scenarios. That is, we will propose a compressive sensing framework, and study uses of it which can produce low-rank factorizations from these measurements where we show that the total number of measurements required scales according to the number of parameters in the factorization, rather than the full, uncompressed tensor. We analyze two recovery approaches along with the necessary specializations of the measurement framework.

Unlike prior works related to algorithms for low-rank tensor approximation from such compressive measurements, we present a unified analysis that permits several different choices for

structured measurement ensembles, and show how to prove error guarantees comparing the error of our recovery algorithm's approximation of the input tensor to the best possible low-rank Tucker approximation error achievable for the tensor by any possible algorithm.

We include empirical and practical studies that verify our theoretical findings and explore various trade-offs of parameters of interest. We discuss the development of the methods, and how theoretical and practical critiques of our earlier work informed and enabled improvements in the sequel.

Next, we consider the related problem of tensor completion, where the goal is to exactly complete a low-rank CP tensor. Our method leverages existing matrix completion techniques and an adaptive sampling scheme along with a noise robust version of Jennrich's Algorithm to complete a tensor using a sample budget which scales, up to a log factor, with the number of parameters in the factorization. Empirical studies, such as performance in the presence of additive noise on simulated data, as well as several practical applications of the method are included and discussed.

Copyright by  
CULLEN HASELBY  
2023

To my father, Ray Haselby.

## ACKNOWLEDGEMENTS

One could try to explore the frontiers alone, but in my view, precious little would be seen this way. Far better, and far more rewarding, to go there with others. This work and what it represents of my understanding of mathematics and its place in the world was accomplished principally through, and because of, discourse and collaboration with others.

The most frequent and immediate target for my graspings for meaning in subjects large and small mathematically is my advisor, Mark Iwen. You taught me much, gave me much, and I never once saw the end of your patience. You invited me into the good and puissant circle of people and ideas you have cultivated, and I was made better for it. I somewhat doubt that your monstrous investment of time and effort into me will ever enrich you personally, but that also never seemed to bother you. Your example is not lost on me. I intend to do what good I can muster with all that was given to me and become worthy of it. Thank you.

Next, I would like to thank Michael Perlmutter, Elizaveta Rebrova, Deanna Needell and William Swartworth for first pulling me onboard, and then rowing along with me on our now shared mission to make tensor recovery with modewise measurements a better understood and more useful art, and with whom I wrote [23] and [24] alongside. Your expertise, curiosity, persistence showed me how the work is done.

Here at MSU, I was buoyed considerably by the fruitful interdisciplinary collaboration with Heiko Hergert, his student Roland Wirth and my graduate student older sibling so-to-speak Ali Zare as seen in our work [68]. I learned more than I think I let on in those early days from reading your writings, pondering your code and methods, and probing your ideas.

Santhosh Karnik with whom I wrote [25]. It was a lot of fun making Tensor Sandwiches with you, and sharpening our ideas together. Thank you.

I would like to thank Jianliang Qian for his service on my committee, his helpful advice when I was taking my first steps into applied mathematics research, his work as an instructor and his direction of the Industrial Mathematics program here at MSU; from which I had very memorable collaboration. I can with the confidence of some hindsight and experience say that numerical linear

algebra and the topics you taught and taught well were used near daily in both my research and work.

To Rongrong Wang, thank you for your service on my committee and your insights into my forays into tensor completion algorithms.

To the faculty, staff, and fellow students at MSU Mathematics department who strove in so many ways to make our collective contribution to human understanding a success, and maintaining a tradition of excellence, I am also indebted. In particular I would like to acknowledge my fellow student Yuta Hozumi. I did value our several hundred hours of pondering each other's many problems and ideas, however it was your friendship that was at times, sustaining. Thank you.

## TABLE OF CONTENTS

CHAPTER 1	INTRODUCTION . . . . .	1
CHAPTER 2	TENSOR RESTRICTED ISOMETRY PROPERTY AND TENSOR ITERATIVE HARD THRESHOLDING FOR RECOVERY . . . . .	35
CHAPTER 3	LEAVE ONE OUT MEASUREMENTS FOR RECOVERY . . . . .	54
CHAPTER 4	TENSOR SANDWICH: TENSOR COMPLETION FOR LOW CP- RANK TENSORS VIA ADAPTIVE RANDOM SAMPLING . . . . .	114
BIBLIOGRAPHY . . . . .		140

# CHAPTER 1

## INTRODUCTION

### 1.1 Introduction

The need for efficient methods to acquire, store, reconstruct, and analyze large-scale data is common now in many settings and disciplines. Furthermore, data may often be multi-modal thus making tensor representations natural and useful in both their storage and analysis. The contexts for this sort of data are numerous and extremely varied, from signal processing tasks, imaging, machine learning, and many other data science applications [37, 12, 59, 42, 9].

Tensors are the natural extension of vectors and matrices, allowing for an arbitrary number of modes, instead of only one or two. Despite seemingly straightforward extensions of the matrix concept, many of the standard linear algebra and computational results that are tried and true in the matrix case simply do not cleanly translate beyond two modes. Nevertheless, tensors of interest rarely have unconstrained, unrelated entries. In practical settings, the tensor data may have some implicit low-rank structure that can be exploited for efficient computation and storage, or simply be well approximated by a tensor that does have this structure. Examples of decompositions include Tucker, CANDECOMP/PARAFAC (CP), and tensor train [37].

Since tensor data can be large both in the number of modes and the dimensions within each mode, and arrive in a streaming fashion, efficient methods that deal with these facets are critical for the computation of such factorizations as well as recovery of the data from compressed measurements or using sampling strategies.

In this work, we will describe and analyze a framework and methods which relate to this framework that can reduce overall memory requirements, theoretically accommodate many types of measurement schemes for collecting linear measurements of large tensors, apply to several low-rank decompositions, while also remaining robust to non-exact low-rankness, and provide computationally tractable and provable recovery. The framework also applies to the streaming setting, such as when the tensor data is being received sequentially over time in a slice by slice, or even entrywise manner. Efficiency in terms of storage for linear measurements are valuable in

streaming algorithms for tensor reconstruction in the big data setting (see, e.g., [60]) since it can be impractical or impossible to have access to the entire, full tensor that one wishes to measure and later reconstruct.

## 1.2 Motivating Examples

The following examples provide context and some indication as to how measurements of tensors and their recovery are of theoretical and practical value.

### 1.2.1 Example, Technical

For example, suppose we wish to aggregate tensor data over time using the additive rule

$$\mathcal{X}_t := \mathcal{X}_{t-1} + \Delta\mathcal{X}_t, \quad \mathcal{X}_0 = 0$$

based on updates  $\Delta\mathcal{X}_t$ . After some number of updates, we wish to reconstruct an estimate of  $\mathcal{X}_T = \sum_{t=1}^T \Delta\mathcal{X}_t \in \mathbb{R}^{n_1 \times \dots \times n_d}$  for some  $T > 0$ , where  $T$  can be large. Storing every intermediate tensor  $\mathcal{X}_t$  in uncompressed form is inefficient. Instead we can store small linear sketches of the intermediate tensors using a linear measurement operator  $\mathcal{L} : \mathbb{R}^{n_1 \times \dots \times n_d} \rightarrow \mathbb{R}^{m_1 \times \dots \times m_{d'}}$  with  $\prod_{\ell=1}^{d'} m_\ell \ll \prod_{\ell=1}^d n_\ell$ . This operator can be used to iteratively update the comparatively smaller sketch

$$\mathcal{L}(\mathcal{X}_t) = \mathcal{L}(\mathcal{X}_{t-1}) + \mathcal{L}(\Delta\mathcal{X}_t)$$

over the updates  $t = 0, \dots, T$ . So long as the final sketch  $\mathcal{L}(\mathcal{X}_T)$  can then be used to approximately recover  $\mathcal{X}_T$  using a recovery algorithm, of sufficient accuracy, then we can achieve storage savings so long as the number of entries in the measurements are smaller than the number of entries in the original signal. Ideally the recovery algorithm itself would also be efficient in terms of memory and computational complexity, so that the cost of recovering  $\mathcal{X}_T$  from  $\mathcal{L}(\mathcal{X}_T)$  does not unduly outweigh the efficiencies in terms of storage of dealing with only the measurements after the operator has acted on the signal.

Naturally, in this situation  $\mathcal{L}$  itself must be stored or newly re-calculated each time it is to operate on a tensor. This is a storage requirement beyond the measurements themselves,  $\mathcal{L}(\mathcal{X}_{t-1})$  and  $\mathcal{L}(\Delta\mathcal{X}_t)$ . Perhaps this operator can be used many times on different streams, yet even so, it is

truly only memory conserving if the cost of storing  $\mathcal{L}$  is less than the savings accrued by not storing the tensor  $\mathcal{X}_t$ , and instead using the measurements, when such is even possible. Indeed, in the classic compressive sensing scenario where one has say a large sensing matrix acting on a sparse vector, one needs to store the measurement operator (or generate it anew each time it is needed) and without a low memory approach or a specially designed measurement operator, the storage of the matrix would exceed any savings gained by the compression of the data. So, by developing our framework, we will by turns describe how to construct a class of operators  $\mathcal{L}$  and pair them with recovery procedures so that for signals of a given type we can with confidence recover them to a desired level of accuracy.

### **1.2.2 Example, Hypothetical and Possibly Non-benign**

Consider the following hypothetical application. Suppose there is a weapon which seeks to replace the role of the landmine in warfare (victim operated mines violate important international treaties and can be indiscriminate, and thus a potentially great evil, even outside of when they have served their initial intended military purpose). Suppose these new weapons have cameras and each device has a limited amount of computational resources, bandwidth, and storage. A well designed measurement operator  $\mathcal{L}$  could enable each device to perform clustering and object recognition on segments of video data. Video data is easily conceived of as a tensor as collected by the devices' cameras. Since the data can be significantly reduced in size while still retaining enough information to perform the needed tasks, this operation can be performed on the device's limited hardware.

However, the machine alone should likely never be able to make the determination that a given segment of video represents a valid target. Once a segment of video data has been identified as significant, these reduced size measurements can be transmitted, and the scene reconstructed so that a human operator can view the scene and make a determination. In this way, one human could potentially supervise many weapons at once, and the system would not be onerous with constant, high volumes of data needing constant transmission back to nodes for computing and storage.

### **1.2.3 Example, Hypothetical and Benign**

Consider another hypothetical. Suppose a firm has designed a method that can reliably identify after some training and tuning, sections of a TV-broadcast which are live sports, and sections of the broadcast which are commercial breaks. Moreover, this procedure relies only on pairwise distance information taken from the streaming video and audio of the broadcast, and the categorization can be accomplished, for appropriately measured (and compressed) data on an inexpensive computer in milliseconds. Given this situation, a large chain of sports bars would be able to deploy the method and accompanying hardware at each of their establishments, across many broadcasts, many sports and many regions, to locally identify whenever an incoming TV broadcast is on commercial break; and automate switching their patron-facing viewscreens to display some other chosen video – for example, the sports bar chain might sell (their own) ads; or play a muted version of the source TV-broadcast. The compressed data can be retained, and recovery performed for quality assurance and possible tuning.

### **1.3 Organization and Contribution**

The rest of this work is organized in the following way. First, in Section 1.4.1, we will lay out definitions, fix notation as well as state some fundamental results in numerical linear algebra and high dimensional probability that will be useful throughout this work. The final Section 1.5 of Chapter 1 will describe the measurement framework as well as outline its development and relationship to other research on these topics.

Chapter 2 will concern what was our first specialization of the framework and how we interfaced it with prior work that concerned a recovery procedure known as Tensor Iterative Hard Thresholding (TIHT) via a concept known as Tensor Restrictive Isometry Property (TRIP). We will state the main results involving TRIP and TIHT in Section 2.1 and describe the theoretical and practical advantages of this specialization as well as limitations in the context of our empirical findings. The numerical experiments as well as the proofs of the main theoretical contents of this chapter as well as the numerical findings first appeared in our work [23]. In that work I was principally responsible for the numerical experiments, comparisons, observations and discussion regarding the memory

footprint, whereas Micahel Perlmutter and Elizaveta Rebrova produced most of the proof for the main results. Since a critique of the main ideas of the work are germane to the development of the sequel, technical proofs for this chapter are put in the Appendix to that chapter, 2.4. The desire to address these critiques of the main findings as described and discussed in Chapter 2 led us to consider a new specialization of the framework, what we coin the Leave-One-Out alternative, for reasons that will be made evident in Chapter 3.

Chapter 3 describes the Leave-One-Out variations of the framework and the paired recovery algorithm. In Section 3.2 we conduct a first of its kind analysis of the method and state and prove recovery guarantees that demonstrate the method has an overall sub-linear dependence on the total size of the tensor data. We also provide experiments on simulated data as well as video data to show how different choices and parameters in the problem contribute to trade-offs of interest. The results of this chapter first appear in our work [24]. In that work I was responsible for both most of the proof writing and numerical experiments. Technical proofs adapted from other research is included in that chapter's Appendix, since these involved important but minor specializations to our task and may distract from the main findings.

Finally, in Chapter 4 we present a novel method for completing tensors that relies on a (mildly) adaptive sampling procedure and prove that in the noiseless setting this method can exactly complete a low-rank tensor. Sampling can be viewed as a very particular type of measurement operator, and though this sampling procedure does not strictly fit into the framework we describe in Section 1.5, its design is very much in the same spirit. We conclude that chapter again with a number of numerical experiments that demonstrate the main findings as well show empirically that the method can succeed in the presence of noise, as well as a number of applications of the method on real world data-sets. The initial results of this chapter appeared in our work [25], the main result, its proof and the empirical findings were primarily my contribution.

We will need to set notation and define common terms and some base results in order to deal with the subject matter.

## 1.4 Preliminaries

### 1.4.1 Tensor and Measurement Preliminaries

**Definition 1.1.** A  $d$ -mode or order- $d$  tensor (or  $d$ -th order) is a  $d$  dimensional array of complex values, written as

$$\mathcal{X} \in \mathbb{C}^{n_1 \times n_2 \times \dots \times n_d}$$

A  $d$ -mode tensor's entries are indexed by a vector  $\mathbf{i} \in [n_1] \times [n_2] \times \dots \times [n_d]$  where  $(\mathcal{X})_{\mathbf{i}} = x_{\mathbf{i}} = x_{i_1, \dots, i_d} \in \mathbb{C}$

**Example 1.1** (1 and 2 mode tensors). (i) A 1-mode tensor,  $\mathbf{x} \in \mathbb{C}^n$  is a vector with entries

$x_i \in \mathbb{C}, i \in [n]$ . We will denote 1-mode tensors, vectors, with bolded lowercase letters.

(ii) A 2-mode tensor,  $A \in \mathbb{C}^{n_1 \times n_2}$  is a matrix with entries  $a_{i_1, i_2} \in \mathbb{C}$  for  $i \in [n_1], i_2 \in [n_2]$ . We denote 2-mode tensors (matrices) with capital letters.

We introduce some terminology that will be useful when describing tensors

**Definition 1.2** (Fiber). Fibers are 1-dimensional subsets of a  $d$ -mode tensor. They are formed by fixing  $d - 1$  of the dimensions and then ranging over all indices in the remaining dimension. So for any  $k \in [d]$ , and  $\mathcal{X} \in \mathbb{C}^{n_1 \times \dots \times n_d}$  then a  $k$ -mode fiber would be a vector  $\mathbf{x} \in \mathbb{C}^{n_k}$  where indices  $i_1, \dots, i_{k-1}, i_{k+1}, \dots, i_d$  are fixed Using Matlab notation

$$(\mathcal{X})_{i_1, \dots, i_{k-1}, :, i_{k+1}, \dots, i_d} = \mathbf{x}_{i_1, \dots, i_{k-1}, :, i_{k+1}, \dots, i_d}$$

So for example, given a matrix  $X \in \mathbb{C}^{n_1 \times n_2}$  then  $X_{i,:} = \mathbf{x}_{i,:} \in \mathbb{C}^{n_2}$  is a mode-2 fiber (i.e. row). A mode-1 fiber,  $\mathbf{x}_{:,j}$  is a column of the matrix.

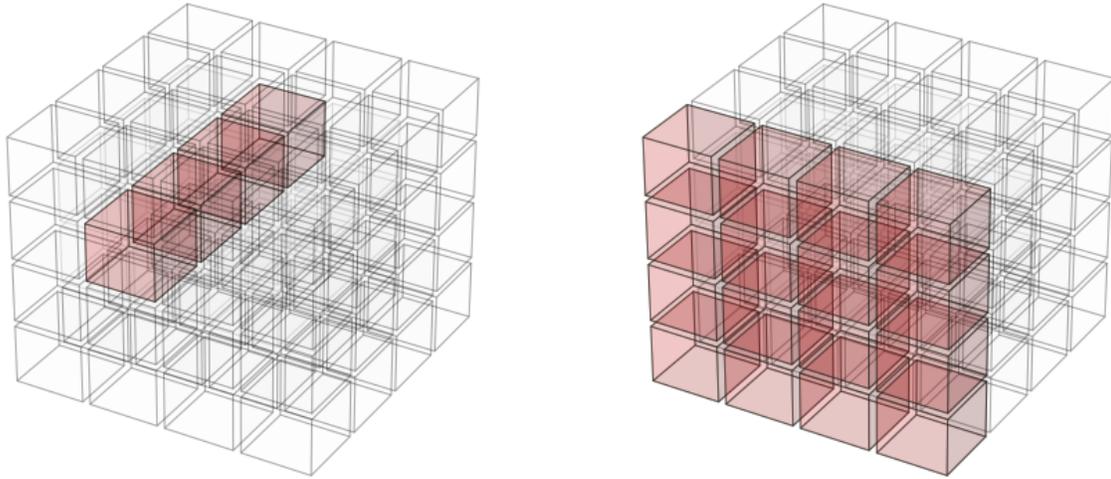
See Figure 1.1 for a schematic depiction of these terms.

**Definition 1.3** (Slice). A matrix formed by varing 2 indices and fixing all other indices of a tensor.

That is, suppose  $j, k \in [d]$  where  $j \neq k$  then

$$X = \mathcal{X}_{i_1, \dots, i_{j-1}, :, i_{j+1}, \dots, i_{k-1}, :, i_{k+1}, \dots, i_d} \in \mathbb{C}^{n_j \times n_k}$$

**Definition 1.4** (Sub-tensor). A  $k$ -mode sub-tensor of a  $d$ -mode tensor ( $k \leq d$ ) is denoted by a vector of length  $d - k$  of indices and a set of  $k$  distinct indices from the set  $[d]$ . That is given



(a) Fiber

(b) Slice

Figure 1.1 A schematic of  $\mathcal{X} \in \mathbb{R}^{4 \times 4 \times 4}$  tensor where the fiber  $\mathbf{x}_{2,2,:}$  is highlighted in (a) and a frontal slice  $X_{:, :, 1}$  is highlighted in b.

distinct  $j_1, \dots, j_k \in [d]$  and vector  $\mathbf{i} \in \bigotimes_{i \neq j_\ell} [n_i]$  of length  $d - k$ , we have a  $k$ -mode sub-tensor where

$$\mathcal{X}_{j_1, \dots, j_k, \mathbf{i}} \in \mathbb{C}^{n_{j_1} \times \dots \times n_{j_k}}$$

Using this sub-tensor notation then an entry of a mode- $k$  fiber is a sub-tensor  $\mathcal{X}_{j, \mathbf{i}}$  where  $j \in [n_k]$  and  $\mathbf{i} \in [n_1] \times \dots \times [n_{k-1}] \times [n_{k+1}] \times \dots \times [n_d]$ . There are  $\prod_{\ell \neq k} n_\ell$  different mode- $k$  fibers, one for each possible  $\mathbf{i}$ .

An entry of a slice then is denoted  $\mathcal{X}_{\ell, k, \mathbf{i}} \in \mathbb{C}^{n_\ell \times n_k}$ . There are  $\prod_{j \neq \ell, k} n_j$  slices of dimension  $n_\ell \times n_k$ .

Next we will discuss reshaping operators - these all involve different possible ways of changing the dimensions and number of modes of tensors so that they have the same number of entries but are different shapes.

**Definition 1.5** (Vectorization). For  $\mathcal{X} \in \mathbb{C}^{n_1 \times \dots \times n_d}$ ,  $\text{vec}(\mathcal{X}) = \mathbf{x}$  where  $\mathbf{x} \in \mathbb{C}^{\prod_{k=1}^d n_k}$ . Entries are typically ordered lexicographically by their indices, i.e. the first index sorts all values before the second and so on. However, usually the particular order that entries are moved to between

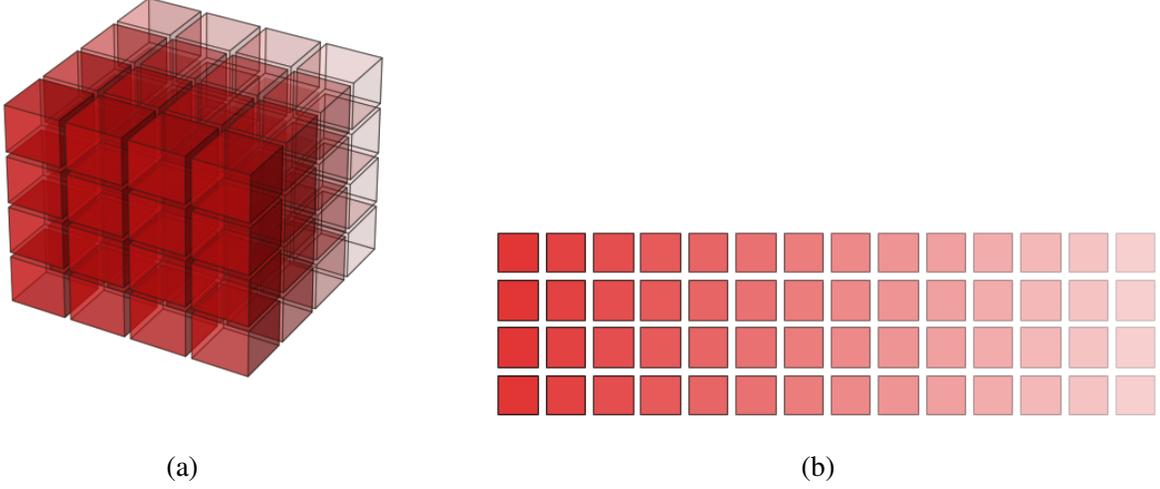


Figure 1.2 A schematic of  $\mathcal{X} \in \mathbb{R}^{4 \times 4 \times 4}$  tensor where the mode-1 fibers  $\mathbf{x}_{:,i,j}$  are arranged as the columns of the corresponding flattening. The intensity of the color corresponds to the lexicographic order of the fibers.

reshapings does not matter, so long as it is consistent across the operations.

**Definition 1.6** (Mode- $k$  Unfolding). Also known as flattening or matricization. For  $\mathcal{X} \in \mathbb{C}^{n_1 \times \dots \times n_d}$  the  $k$ -mode flattening is a matrix  $X_{(k)} \in \mathbb{C}^{n_k \times \prod_{j \neq k} n_j}$ . We have effectively made the  $k$ -th dimension into the rows of the matrix, and the columns are then the different mode- $k$  fibers, i.e. the columns are the fibers  $\mathcal{X}_{k,i}$ . When it will not cause any confusion from context, we may omit the decoration  $(k)$  and simply write the flattening as  $X$ .

**Definition 1.7** (Reshaping). We can reshape a  $d$ -mode tensor into any other  $f$ -mode tensor with a reshaping operation  $R : \mathbb{C}^{n_1 \times \dots \times n_d} \rightarrow \mathbb{C}^{m_1 \times \dots \times m_f}$  provided

$$\prod_{j=1}^d n_j = \prod_{\ell=1}^f m_\ell$$

$k$ -mode flattening and vectorization are in this view two particular reshaping operations.

What is the underlying vector space we can use to study tensors? To answer that, we consider the following norm, inner-product, and operations on tensors:

**Definition 1.8** (2-norm of a Tensor). For  $\mathcal{X} \in \mathbb{C}^{n_1 \times \dots \times n_d}$  then given any  $k \in [d]$  we have

$$\|\mathcal{X}\|_2^2 = \|X_{(k)}\|_F^2 = \|\mathbf{x}\|_2^2 = \sum_{\mathbf{i} \in I} |x_{\mathbf{i}}|^2$$

where  $I = [n_1] \times \dots \times [n_d]$

**Definition 1.9** (Inner-product). For compatible tensors  $\mathcal{X}, \mathcal{Y} \in \mathbb{C}^{n_1 \times \dots \times n_d}$  then

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{\mathbf{i} \in I} x_{\mathbf{i}} \bar{y}_{\mathbf{i}} = \langle \mathbf{x}, \mathbf{y} \rangle$$

that is, the inner-product of the vectorization of the tensors. Note that this is equivalent to  $\langle X_{(k)}, Y_{(k)} \rangle_{HS} = \text{Trace}(X_{(k)}(Y_{(k)})^*)$ ,  $\forall k \in [d]$ , the Hilbert-Schmidt inner product for matrices

Addition and scalar multiplication work component-wise, i.e.

$$(\mathcal{X} + \mathcal{Y})_{\mathbf{i}} = x_{\mathbf{i}} + y_{\mathbf{i}}$$

$$(\alpha \mathcal{X})_{\mathbf{i}} = \alpha x_{\mathbf{i}}, \forall \alpha \in \mathbb{C}$$

## 1.4.2 Modewise and Matrix Products

The modewise product will feature prominently in our introduction to the measurement framework Section 1.5, as well as Chapters 2 and 3. We state it here as well as some basic algebraic identities related to it. Certain matrix products will also quickly become useful when describing tensors and their decompositions.

**Definition 1.10** (Modewise Product). The  $k$ -mode product of  $\mathcal{X} \in \mathbb{C}^{n_1 \times \dots \times n_d}$  and  $U \in \mathbb{C}^{m_k \times n_k}$  for  $k \in [d]$  is a tensor in  $\mathbb{C}^{n_1 \times \dots \times n_{k-1} \times m_k \times n_{k+1} \times \dots \times n_d}$  element-wise defined as

$$(\mathcal{X} \times_k U)_{i_1, \dots, i_{k-1}, j, i_{k+1}, \dots, i_d} = \sum_{i_k=1}^{n_k} x_{i_1, i_2, \dots, i_d} u_{j i_k} \quad (1.1)$$

This is usefully identified with the following vectorized expression

$$(\mathcal{X} \times_k U)_{i_1, \dots, i_{k-1}, :, i_{k+1}, \dots, i_d} = U \mathcal{X}_{i_1, \dots, i_{k-1}, :, i_{k+1}, \dots, i_d}$$

In other words, the  $k$ -mode product applies the matrix  $U$  to all the mode- $k$  fibers of the tensor  $\mathcal{X}$ . For example, suppose  $\mathcal{X} \in \mathbb{C}^{5 \times 3 \times 2}$  and  $U \in \mathbb{C}^{4 \times 5}$ . Then  $\mathcal{X} \times_1 U \in \mathbb{C}^{4 \times 3 \times 2}$  where each of the mode-1 fibers is now the product of  $U \mathcal{X}_{:, j, \ell}$ , for some  $j \in [3], \ell \in [2]$ .

In the 2-mode tensor case, i.e., matrices, the usual matrix-matrix multiplication can be understood in terms of 1-mode tensor product.

This also easily establishes that we can identify a mode-wise product with the matrix-matrix multiplication of the matrix with the corresponding  $k$ -mode flattening of the tensor.

$$\begin{aligned} AB &= A \left[ \mathbf{b}_1 \mid \mathbf{b}_2 \mid \dots \mid \mathbf{b}_n \right] \\ &= \left[ A\mathbf{b}_1 \mid A\mathbf{b}_2 \mid \dots \mid A\mathbf{b}_n \right] \\ &= B \times_1 A \end{aligned}$$

Note, modewise products are bilinear. We note some other useful properties.

**Lemma 1.1** (Properties of  $k$ -mode products). *Let  $\mathcal{X}, \mathcal{Y} \in \mathbb{C}^{n_1, \dots, n_d}$ ,  $\alpha, \beta \in \mathbb{C}$ ,  $U_\ell, V_\ell \in \mathbb{C}^{m_\ell \times n_\ell}$ ,  $\forall \ell \in [d]$ . Then*

1.  $(\alpha\mathcal{X} + \beta\mathcal{Y}) \times_j U_j = \alpha(\mathcal{X} \times_j U_j) + \beta(\mathcal{Y} \times_j U_j)$
2. *that is,  $k$ -mode product is bilinear.*
3. *If  $j \neq \ell$  then*

$$(\mathcal{X} \times_j U_j) \times_\ell V_\ell = (\mathcal{X} \times_\ell V_\ell) \times_j U_j$$

4. *If  $W \in \mathbb{C}^{p \times m_j}$  then  $(\mathcal{X} \times_j U_j) \times_j W = \mathcal{X} \times_j (WU_j) \in \mathbb{C}^{n_1 \times n_{j-1} \times p \times n_{j+1} \times \dots \times n_d}$*

*Proof.* 1. For any  $i_1 \in n_1, \dots, i_{j-1} \in n_{j-1}, i_{j+1} \in n_{j+1}, \dots, i_d \in n_d$  we have

$$\begin{aligned} [(\alpha\mathcal{X} + \beta\mathcal{Y}) \times_j U]_{i_1, \dots, i_{j-1}, i_{j+1}, \dots, i_d} &= U(\alpha\mathcal{X} + \beta\mathcal{Y})_{i_1, \dots, i_{j-1}, i_{j+1}, \dots, i_d} \\ &= \alpha U\mathcal{X}_{i_1, \dots, i_{j-1}, i_{j+1}, \dots, i_d} + \beta U\mathcal{Y}_{i_1, \dots, i_{j-1}, i_{j+1}, \dots, i_d} \\ &= (\mathcal{X} \times_j U + \beta(\mathcal{Y} \times_j U))_{i_1, \dots, i_{j-1}, i_{j+1}, \dots, i_d} \end{aligned}$$

2. Similar to 1.

3. Consider the element-wise definition (1.1)

$$\begin{aligned} ((\mathcal{X} \times_j U_j) \times_\ell V_\ell)_{i_1, \dots, p, i_{j+1}, \dots, q, i_{\ell+1}, \dots, i_d} &= \sum_{i_q=1}^{n_q} \sum_{i_p=1}^{n_p} x_{i_1, i_2, \dots, i_d} u_{p, i_p} v_{q, i_q} \\ &= \sum_{i_p=1}^{n_p} \sum_{i_q=1}^{n_q} x_{i_1, i_2, \dots, i_d} u_{p, i_p} v_{q, i_q} \\ &= ((\mathcal{X} \times_\ell V_\ell) \times_j U_j)_{i_1, \dots, p, i_{j+1}, \dots, q, i_{\ell+1}, \dots, i_d} \end{aligned}$$

4. Using the same identity as in 3, the proof is similar

□

Note that the run-time complexity is the same regardless of the order one applies the  $k$ - or  $j$ -mode products.

**Definition 1.11** (Kronecker Product). The Kronecker product of two matrices  $A \in \mathbb{C}^{m_1 \times n_1}$  and  $B \in \mathbb{C}^{m_2 \times n_2}$  is a matrix

$$A \otimes B := \begin{bmatrix} a_{1,1}B & a_{1,2}B & \dots & a_{1,n_1}B \\ a_{2,1}B & a_{2,2}B & \dots & a_{2,n_1}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m_1,1}B & a_{m_1,2}B & \dots & a_{m_1,n_1}B \end{bmatrix}. \quad (1.2)$$

where  $A \otimes B \in \mathbb{C}^{m_1 m_2 \times n_1 n_2}$

**Definition 1.12** (Khatri-Rao Product). The Khatri-Rao product of two matrices is the matrix that results from computing the Kronecker product of their matching columns. That is, for  $A \in \mathbb{R}^{m_1 \times n}$  and  $B \in \mathbb{R}^{m_2 \times n}$ , their Khatri-Rao product is the matrix  $A \odot B \in \mathbb{R}^{m_1 m_2 \times n}$  defined by

$$A \odot B := \begin{bmatrix} \mathbf{a}_1 \otimes \mathbf{b}_1 & \mathbf{a}_2 \otimes \mathbf{b}_2 & \dots & \mathbf{a}_n \otimes \mathbf{b}_n \end{bmatrix}.$$

We also use a so-called row-wise Khatri-Rao product (sometimes called the *face-splitting product*) of  $A \in \mathbb{R}^{m \times n_1}$  and  $B \in \mathbb{R}^{m \times n_2}$ , denoted by  $A \cdot B \in \mathbb{R}^{m \times n_1 n_2}$ , and defined by  $(A \cdot B)^T := A^T \odot B^T$ .

**Lemma 1.2.** Let  $\mathcal{X} \in \mathbb{C}^{n_1, \dots, I_d}$ ,  $U_\ell \in \mathbb{C}^{m_\ell \times n_\ell}$  then

1.  $(\mathcal{X} \times_j U_j)_{(j)} = U_j \mathcal{X}_{(j)} \in \mathbb{C}^{m_j \times \prod_{\ell \neq j} n_\ell}$
2.  $(\mathcal{X} \times_1 U_1 \times_2 U_2 \cdots \times_d U_d)_{[j]} = U_j \mathcal{X}_{[j]} (U_d \otimes U_d \otimes \cdots \otimes U_{j+1} \otimes U_{j-1} \otimes \cdots \otimes U_1)^T$

*Proof.* This result follows from the definition of reshaping operators, which we detail below, assuming a column-major convention, and by using the elementwise definition in (1.1). □

In order for the identity seen in Lemma 1.2 to hold, we need specify our precise matricization convention. In our convention, the entry at location  $(i_1, \dots, i_d)$  in  $\mathcal{X} \in \mathbb{C}^{n_1 \times \cdots \times n_d}$  is located in the

matrix as entry  $(i_p, j)$  where

$$j = 1 + \sum_{\substack{k=1 \\ k \neq p}}^d (i_k - 1) J_k$$

where

$$J_k = \prod_{\substack{m=1 \\ m \neq p}}^{k-1} n_m$$

set  $J_k = 1$  if the index of the product above is empty. These conventions can become cumbersome to state, however they are made clear and more natural by an example.

**Example 1.2** (3-mode). A similar example appears in [37]: Consider a tensor  $\mathcal{X} \in \mathbb{R}^{3 \times 4 \times 2}$ , the frontal slices are as follows

$$\mathcal{X}_{::,1} = \begin{pmatrix} 1 & 4 & 7 & 10 \\ 2 & 5 & 8 & 11 \\ 3 & 6 & 9 & 12 \end{pmatrix}, \mathcal{X}_{::,2} = \begin{pmatrix} 13 & 16 & 19 & 22 \\ 14 & 17 & 20 & 23 \\ 15 & 18 & 21 & 24 \end{pmatrix}$$

The three different unfoldings are then as follows. Consider  $p = 1$ ,

$$X_{(1)} = \begin{pmatrix} 1 & 4 & 7 & 10 & 13 & 16 & 19 & 22 \\ 2 & 5 & 8 & 11 & 14 & 17 & 20 & 23 \\ 3 & 6 & 9 & 12 & 15 & 18 & 21 & 24 \end{pmatrix}$$

Note

$$J_1 = \prod_{\substack{m=1 \\ m \neq 1}}^{1-1} (n_m) = 1, J_2 = \prod_{\substack{m=1 \\ m \neq 1}}^{2-1} n_m = 1, J_3 = \prod_{\substack{m=1 \\ m \neq 1}}^{3-1} n_m = 4$$

Now to see how to locate a particular entry, note that  $\mathcal{X}_{2,3,2} = 20$ , so in our unfolding  $(\mathcal{X}_{(p)})_{(i,j)}$  we can simply copy the index that corresponds to the  $p$ -th mode, i.e. the first here,  $i = 2$ . To find the column, compute  $j = 1 + \sum_{\substack{k=1 \\ k \neq p}}^d (i_k - 1) J_k = 1 + (i_2 - 1) J_2 + (i_3 - 1) J_3 = 1 + (2)(1) + (1)(4) = 7$ . So our entry with value 20 is in location  $(2, 7)$ .

Consider  $p = 2$ ,

$$\mathcal{X}^{(1)} = \begin{pmatrix} 1 & 2 & 3 & 13 & 14 & 15 \\ 4 & 5 & 6 & 16 & 17 & 18 \\ 7 & 8 & 9 & 19 & 20 & 21 \\ 10 & 11 & 12 & 22 & 23 & 24 \end{pmatrix}$$

Again, to locate our entry,  $\mathcal{X}_{2,3,2} = 20$ , we return the index on the  $p$ -th mode as our row,  $i = 3$ , and repeat the same calculation for  $j$ :

Note

$$J_1 = \prod_{\substack{m=1 \\ m \neq 2}}^{1-1} (n_m) = 1, J_2 = \prod_{\substack{m=1 \\ m \neq 2}}^{2-1} n_m = 1, J_3 = \prod_{\substack{m=1 \\ m \neq 2}}^{3-1} n_m = 3$$

This time we leave out the  $J_2$  factor:  $j = 1 + \sum_{\substack{k=1 \\ k \neq n}}^{d-1} (i_k - 1)J_k = 1 + (1)(1) + (1)(3) = 5$ . So our entry with value 20 is located in (3, 5).

Consider  $p = 3$ ,  $\mathcal{X}_{(3)}$

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & \dots \\ 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & \dots \end{pmatrix}$$

Again, to locate our entry,  $\mathcal{X}_{2,3,2} = 20$ , we return the index on the  $p$ -th mode as our row,  $i = 1$ , and repeat the same calculation for  $j$ :

Note

$$J_1 = \prod_{\substack{m=1 \\ m \neq 3}}^{1-1} (n_m) = 1, J_2 = \prod_{\substack{m=1 \\ m \neq 3}}^{2-1} n_m = 3, J_3 = \prod_{\substack{m=1 \\ m \neq 3}}^{3-1} n_m = 12$$

This time we leave out the  $J_3$  factor:  $j = 1 + \sum_{\substack{k=1 \\ k \neq n}}^d (i_k - 1)J_k = 1 + 1(1) + 2(3) = 8$ . So our entry with value 20 is located in (2, 8).

### 1.4.3 Decomposition and Low Rank Approximation

Our next set of definitions concerns ways of factorizing tensors. In the exact case, these describe how we can compute or represent the tensor as a product and sum of appropriately constrained sub-tensors of various types. This is useful for many reasons, such as interpreting the data, storing it, or performing computations more efficiently. Approximation will also be our concern. That

is, for situations where exact factorization is not required (or even possible) we wish to find and understand methods that can be used to find approximate tensors that do admit a certain factorization and understand the trade-offs such as how efficiently it is to compute or how accurately we can expect it to resemble the original tensor. To motivate why decompositions are of interest, and provide an example for why there is a need for tensor specific decompositions, we include an example of the storage benefit achieved by applying a standard linear algebra strategy. By tensor specific, we mean doing something other than reshaping tensors into familiar objects (vectors and matrices) and reusing the concepts from basic linear algebra.

Suppose we have  $q$  tensors of size  $\mathbb{C}^{n_1 \times \dots \times n_d}$ ,  $\mathcal{X}_1, \dots, \mathcal{X}_q$ . We will compress, or reduce the dimension, by performing Principal Component Analysis (PCA) on the vectorized tensors. Our goal then in this case is to solve the minimization problem. Assuming the vectorized tensors have been centered, this amounts to

$$\min_{\substack{Q \in \mathbb{R}^{\prod_{k=1}^d n_k \times m} \\ Q^H Q = I}} \sum_{j=1}^q \|\mathbf{x}_j - Q Q^H \mathbf{x}_j\|_2$$

where  $\text{vec}(\mathcal{X}_j) = \mathbf{x}_j \in \mathbb{C}^{\prod_{k=1}^d n_k}$ . We can use the Singular Value Decomposition (SVD) of the following data matrix

$$\left[ \begin{array}{c|c|c|c} \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_q \end{array} \right] = U \Sigma V^* \in \mathbb{C}^{q \times \prod_{k=1}^d n_k}$$

As is well known, and typical for computing projections using PCA, this amounts to storing and using the truncated SVD of the data matrix. Once we have the (top) singular vectors, we can tensorize their outer product using the inverse of our vectorizing reshaping operation. That is

$$\mathcal{X}_j \approx \sum_{k=1}^m \sigma_{j,k} \mathcal{T}_k$$

where  $\mathcal{T}_k$  are the tensors obtained from taking the relevant indices from the sum of outer product of the singular vectors in  $U$  and  $V$ , scaled by the singular vectors.

What does this achieve in terms of storage? The space required for our original collection of tensors  $\mathcal{X}_1, \dots, \mathcal{X}_q \in \mathbb{C}^{n_1 \times \dots \times n_d}$  is  $\mathcal{O}(q \prod_{k=1}^d n_k)$ . After PCA, we need keep  $m$  basis tensors of the

same dimension  $\mathbb{C}^{n_1 \times \dots \times n_d}$  and our coordinates or principal scores will also need to be stored and there are  $m^d$  of these. So the space is  $O(m^d + m \prod_{k=1}^d n_k)$  which may be unsatisfactory because the dependence on  $\prod_{k=1}^d n_k$  is unchanged. Additionally, there's no obvious interpretable structure to the basis tensors  $\mathcal{T}_k$  that relates to the fact the original data was multi-modal. This motivates us then to look to another approach for decomposing (and therefore saving on storage among other things) tensors.

**Definition 1.13** (Rank one Tensor). Given  $d$ -vectors  $\mathbf{x}_j \in \mathbb{C}^{n_j}$  for  $j \in [d]$ , the outer-product

$$\mathcal{X} = \bigcirc_{j=1}^d \mathbf{x}_j \in \mathbb{C}^{n_1 \times \dots \times n_d}$$

has entries given the product of corresponding entries of the vectors, i.e.

$$\mathcal{X}_{i_1, \dots, i_d} = \left( \bigcirc_{j=1}^d \mathbf{x}_j \right)_{i_1, \dots, i_d} = (\mathbf{x}_1)_{i_1} (\mathbf{x}_2)_{i_2} \dots (\mathbf{x}_d)_{i_d}$$

any  $d$ -mode tensor where it is possible to write it as such an outer product of  $d$  vectors is a rank one tensor.

Note that storing a rank one tensor can be accomplished by storing only the vector components, rather than all entries, and computing the entries (as required) using the factors. This definition in the 2-mode case is the familiar rank one matrix case, for  $\mathbf{u} \in \mathbb{C}^m, \mathbf{v} \in \mathbb{C}^N$

$$A = \mathbf{u} \circ \mathbf{v} = \mathbf{u} \mathbf{v}^*$$

then matrix  $A \in \mathbb{C}^{m \times N}$  is a rank one matrix.

**Definition 1.14** (CP decomposition). We say  $\mathcal{X} \in \mathbb{C}^{n_1 \times \dots \times n_d}$  is a CP rank- $r$  tensor if it can be written as the sum of minimally  $r$  rank one tensors. That is

$$\mathcal{X} = \sum_{\ell=1}^r \left( \bigcirc_{j=1}^d \mathbf{a}_j^{(\ell)} \right) = \sum_{\ell=1}^r \mathbf{a}_1^{(\ell)} \circ \dots \circ \mathbf{a}_d^{(\ell)}$$

for vectors  $\mathbf{a}_k^{(\ell)} \in \mathbb{R}^{n_k}$  for  $k \in [d]$  and  $\ell \in [r]$ . For convenience, define factor matrices  $A^{(k)} \in \mathbb{R}^{n_k \times r}$  by  $A^{(k)} = \begin{bmatrix} \mathbf{a}_k^{(1)} & \dots & \mathbf{a}_k^{(r)} \end{bmatrix}$  for  $k \in [d]$  so that the entries of the tensor are given by

$$\mathcal{X}_{i_1, \dots, i_d} = \sum_{\ell=1}^r \prod_{k=1}^d A_{i_k, \ell}^{(k)}$$

At times it will be convenient to require the CP decomposition's component vectors to be normalized to have norm of one, and then weight correspondingly each rank one term with a scalar. That is,

$$\mathcal{X} = \sum_{\ell=1}^r \left( \bigcirc_{j=1}^d \alpha_j \mathbf{a}_j^{(\ell)} \right)$$

where  $\left\| \mathbf{a}_j^{(\ell)} \right\|_2 = 1$  for all  $j \in [d], \ell \in [r]$ . and  $\alpha_j \in \mathbb{C}$

Here we also include a Lemma which relates how modewise products interact with rank-1 components.

**Lemma 1.3.** *Suppose we have a rank one tensor  $\mathcal{X} \in \mathbb{C}^{n_1 \times \dots \times n_d}$  with components  $\mathbf{x}_j \in \mathbb{C}^{n_j}$  for all  $j \in [d]$ , and suppose  $U_k \in \mathbb{C}^{m_k \times n_k}$  for some  $k \in [d]$  then*

$$\left( \bigcirc_{j=1}^d \mathbf{x}_j \right) \times_k U_k = \left( \bigcirc_{j=1}^{k-1} \mathbf{x}_j \right) \circ U_k \mathbf{x}_k \circ \left( \bigcirc_{j=k+1}^d \mathbf{x}_j \right)$$

*Proof.* Note that the  $k$ -mode fibers the tensor  $\left( \bigcirc_{j=1}^d \mathbf{x}_j \right)$  are scalar multiples of the same vector,  $\mathbf{x}_k$

That is, the  $k$ -mode fiber indexed by  $(\ell_0, \dots, \ell_{k-1}, \ell_{k+1}, \dots, \ell_d)$  is

$$\left( \prod_{j \neq k}^{d-1} (\mathbf{x}_j)_{\ell_j} \right) \mathbf{x}_k$$

but  $\left( \prod_{j \neq k}^d (\mathbf{x}_j)_{\ell_j} \right)$  is a scalar. So the identity follows now from noting the definition of the mode- $k$  product. (Each column of the unfolding is a scalar multiple of the same vector, scalar commutes with matrix-vector multiplication)  $\square$

As a remark, many of the correspondences and equivalences with rank in the matrix case do not extend to tensors. For example, the rank of a matrix is always equal to the dimension of the space spanned by its columns, and also the dimension of the space spanned by its rows. However, consider the following three mode tensor:

$$\mathcal{X}_{:, :, 1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}, \quad \mathcal{X}_{:, :, 2} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Naturally, the four mode-1 fibers span a four dimensional space, however the eight different mode-2 fibers span a two dimensional space. In general there is no simple correspondence between the dimensions of the spaces spanned by the different sets of fibers for a tensor.

There are other complications that arise with notions of rank in the case of higher order tensors. In the case of matrices, a rank 3 matrix has a unique, optimal rank 2 approximation, which can be obtained from truncating its SVD. In the case of tensors, it is entirely possible for a rank 3 tensor to be approximated arbitrarily close by a rank 2 tensor, as this simple example below demonstrates:

$$\mathcal{X}_{::,1} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \mathcal{X}_{::,2} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$$

Now consider

$$\mathcal{A}_{::,1} = \begin{pmatrix} \epsilon^{-1} & 1 \\ 1 & \epsilon \end{pmatrix}, \mathcal{A}_{::,2} = \begin{pmatrix} 1 & \epsilon \\ \epsilon & \epsilon^{-2} \end{pmatrix}; \quad \mathcal{B}_{::,1} = \begin{pmatrix} -\epsilon^{-1} & 0 \\ 0 & 0 \end{pmatrix}, \mathcal{B}_{::,2} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

Both  $\mathcal{A}$  and  $\mathcal{B}$  have rank one, so their sum has at most rank two, but we can let  $\mathcal{A} + \mathcal{B}$  get within any distance of  $\mathcal{X}$  by sending  $\epsilon \rightarrow 0$ ; so in this instance, there is no best rank 2 approximation of the rank 3 tensor  $\mathcal{X}$ . Indeed, even if there is no issue between so called border rank of a tensor and its exact rank, unlike the case of a matrix, there is not necessarily any straightforward relationship between the best  $r - 1$  approximation to that tensor and its exact rank  $r$  decomposition in the same way as the SVD and the truncated SVD are related in the case of matrices.

### 1.4.3.1 The Tucker Decomposition and Low-Rank Approximation

Next, we define the Tucker decomposition of tensors, which decomposes a tensor into a core tensor that is multiplied using the modewise product by a factor matrix along each of its modes.

**Definition 1.15** (Tucker decomposition). Let  $\mathcal{X} \in \mathbb{R}^{n_1 \times \dots \times n_d}$ . We say that  $\mathcal{X}$  has a Tucker decomposition of rank  $\mathbf{r} = (r_1, \dots, r_d)$  if there exists  $\mathcal{G} \in \mathbb{R}^{r_1 \times \dots \times r_d}$ , and  $U_j \in \mathbb{R}^{n_j \times r_j}$  with orthonormal columns for all  $j \in [d]$ , so that

$$\mathcal{X} = \mathcal{G} \times_1 U_1 \times_2 U_2 \times_3 \dots \times_d U_d = \sum_{\mathbf{i} \in \{(i_1, \dots, i_d) \mid i_j \in [r_j] \ \forall j \in [d]\}} \mathcal{G}_{\mathbf{i}} \cdot \mathbf{u}_{i_1}^{(1)} \circ \mathbf{u}_{i_2}^{(2)} \circ \dots \circ \mathbf{u}_{i_d}^{(d)}, \quad (1.3)$$

where  $\mathbf{u}_{i_j}^{(j)}$  denotes the  $i_j$ -th column of  $U_j$ , and  $\mathcal{G}_i := \mathcal{G}_{i_1, \dots, i_d} \in \mathbb{R}$ . We will also use the shorthand  $\mathcal{X} = [[\mathcal{G}, U_1, \dots, U_d]] := \mathcal{G} \times_1 U_1 \times_2 U_2 \times_3 \dots \times_d U_d$  when referring to the rank- $\mathbf{r}$  Tucker decomposition of a given rank- $\mathbf{r}$  tensor  $\mathcal{X}$ .

There is an equivalent formulation to equation (1.3) using unfoldings of  $\mathcal{X}$  and of  $\mathcal{G}$  in terms of matrix-matrix products:

$$X_{[j]} = U_j G_{[j]} (U_d \otimes \dots \otimes U_{j+1} \otimes U_{j-1} \otimes \dots \otimes U_1)^T \quad (1.4)$$

From this definition several important observations can be made. First, we see that every tensor must admit a Tucker decomposition, since setting the core tensor equal to the full tensor and the factors equal to the appropriately sized identity matrices would ensure the decomposition matched the full tensor exactly, and the rank would in that case be the same as the length of the modes,  $\mathbf{r} = (n_1, \dots, n_d)$ . Naturally when considering efficient ways to store and perform computations with tensors, we will be interested in the cases where the rank along at least one or more modes is less than the original tensor's side length. And just as in the case of the CP decomposition, what the minimal Tucker rank is and what its relationship is to the exact decomposition is by no means straightforward generically. Second, we can see that the decomposition has no chance of being unique; consider any non-singular matrix  $Y$ , and observe:

$$\begin{aligned} X_{[j]} &= U_j G_{[j]} (U_d \otimes \dots \otimes U_{j+1} \otimes U_{j-1} \otimes \dots \otimes U_1)^T \\ &= U_j Y Y^{-1} G_{[j]} (U_d \otimes \dots \otimes U_{j+1} \otimes U_{j-1} \otimes \dots \otimes U_1)^T \\ &= (U_j Y) (Y^{-1} G_{[j]}) (U_d \otimes \dots \otimes U_{j+1} \otimes U_{j-1} \otimes \dots \otimes U_1)^T \end{aligned}$$

and for this same reason, it is often natural to insist that the factor matrices,  $U_i$  have orthonormal columns, given a Tucker decomposition. We can always arrive at this convention by means of non-singular transformations in each of the modes, e.g. perform a  $QR$  decomposition and have  $R$  act on the core.

There is a specific type of Tucker decomposition that will be of interest, and that is the Higher Order SVD (HOSVD). In addition to the convention of having the columns of the factor matrices

form an orthonormal basis, in the case of the HOSVD, we impose an additional constraint that all the  $(d - 1)$ -mode sub-tensors of the core formed by fixing an index along the same mode are orthogonal.

**Definition 1.16.** [Higher Order Singular Value Decomposition (HOSVD)]. The Tucker decomposition of tensor  $\mathcal{X} = [[\mathcal{G}, U_1, \dots, U_d]]$  is a HOSVD with rank  $\mathbf{r}$ , when for every  $\mathbf{i}_a = (:, :, \dots, a, \dots, :)$ ,  $\mathbf{i}_b = (:, :, \dots, b, \dots, :)$ ,  $a, b \in [r_j]$ ,  $a \neq b$ ,  $j \in [d]$  we have that

$$\langle \mathcal{G}_{\mathbf{i}_a}, \mathcal{G}_{\mathbf{i}_b} \rangle = 0 \quad (1.5)$$

and  $U_k$  have orthonormal columns for all  $k \in [d]$ .

See [15] for a thorough description of this particular type of Tucker decomposition, and many of its useful properties. Because of its straightforward way to compute and often good approximation qualities, the HOSVD is frequently used as an initialization for finding Tucker decompositions.

A HOSVD of a tensor can be computed using Algorithm 1.1.

---

**Algorithm 1.1** Higher Order SVD

---

**input :**

$\mathcal{X} \in \mathbb{C}^{n_1 \times \dots \times n_d}$

$\mathbf{r} = (r_1, r_2, \dots, r_d)$  the rank of the HOSVD

**output :**  $\hat{\mathcal{X}} = [[\mathcal{G}, U_1, \dots, U_d]]$

**for**  $i \in [d]$  **do**

    # SVDs of flattened tensor

    Compute SVD of mode- $j$  unfolding  $X_{(j)} = Q_j \Sigma_j V_j^*$

$U_j \leftarrow Q_j^{(i)}[:, : r_j]$

**end**

# Compute Core

$\mathcal{G} \leftarrow \mathcal{X} \times_{j=1}^d U_j^* \in \mathbb{C}^{r_1 \times \dots \times r_d}$

---

Note that for each unfolding, the full SVD has form

$$A^{(j)} = \underbrace{U}_{n_j \times n_j} \underbrace{\Sigma}_{n_j \times \prod_{j \neq k} n_k}_{n_k} \underbrace{V^H}_{\prod_{j \neq k} n_k \times \prod_{j \neq k} n_k}$$

the  $r_j$ -truncated SVD has form

$$A^{(j)} \approx \underbrace{U}_{n_j \times r_j} \underbrace{\Sigma}_{r_j \times r_j}_{r_j} \underbrace{V^H}_{r_j \times \prod_{j \neq k} n_k}$$

This problem then is repeated for each of the  $d$  different modes. This is potentially a bottleneck computationally and so can likely benefit from fast approximations to the SVD, e.g. see [48].

In order to further improve the decomposition, we can take an alternating least squares approach, successively solving for  $U_j$  while holding other modes constant and iterate on this processes. This is called Higher Order Orthogonal Iteration (HOOI), Algorithm 1.2, see [16]

---

**Algorithm 1.2** Higher Order Orthogonal Iteration

---

**input :**

$$\mathcal{X} \in \mathbb{C}^{n_1 \times \dots \times n_d}$$

$$\mathbf{r} = (r_1, r_2, \dots, r_d)$$

$M$  the maximum number of iterations

**output :**  $\hat{\mathcal{X}} = [[\mathcal{G}, U_1, \dots, U_d]]$

$[[\mathcal{G}^{(0)}, U_1^{(0)}, \dots, U_d^{(0)}]] \leftarrow \text{HOSVD}(\mathcal{X}, \mathbf{r})$  **for**  $i \in [M]$  **do**

**for**  $j \in [d]$  **do**

        # Update each factor matrix

$$\text{Compute } \left( X \times_{k \neq j} \left( U_k^{(i-1)} \right)^* \right)^{(j)} = Q_j^{(i)} \Sigma_j^{(i)} \left( V^{(i)} \right)_j^*$$

$$U_j \leftarrow Q_j^{(i)}[:, : r_j]$$

**end**

**end**

# Compute Core

$$\mathcal{G} \leftarrow X \times_{k \neq j} \left( U_k^{(M)} \right)^*$$


---

We also observe that there is a natural correspondence between CP and Tucker decompositions. Given a CP decomposition, one can form a Tucker decomposition by arranging the weights of the CP decomposition along the diagonal of the core tensor and placing the component vectors as the columns of the factor matrices, while setting all off diagonal entries in the core equal to zero. The following small result formalizes this correspondence.

**Lemma 1.4.** *If  $\mathcal{X} \in \mathbb{C}^{n_1 \times \dots \times n_d}$  has Tucker rank  $(r_1, \dots, r_d)$  then it has CP rank of at most  $\prod_{j=1}^d r_j$*

*Proof.* The tensor has an exact Tucker decomposition, so

$$\mathcal{X} = \mathcal{G} \times_{j=1}^d U_j$$

Note that we can express any tensor in the standard basis; here the standard basis for tensors is a tensor with only one non-zero entry.

So for example for some  $\ell \in [r_1] \times \cdots \times [r_d]$  the associated standard basis element is

$$\bigcirc_{j=1}^d \mathbf{e}_{\ell_j}$$

where  $\mathbf{e}_{\ell_j}$  is the usual standard basis vector in  $\mathbb{C}^{r_j}$ . Denote  $\mathcal{I} = [r_1] \times \cdots \times [r_d]$ . Thus

$$C = \sum_{\ell \in \mathcal{I}} \mathcal{G}_\ell \left( \bigcirc_{j=1}^d \mathbf{e}_{\ell_j} \right)$$

Now use this expression in the Tucker decomposition of  $\mathcal{X}$

$$\begin{aligned} \mathcal{X} &= \mathcal{G} \times_{j=1}^d U_j \\ &= \left[ \sum_{\ell \in \mathcal{I}} \mathcal{G}_\ell \left( \bigcirc_{j=1}^d \mathbf{e}_{\ell_j} \right) \right] \times_{j=1}^d U_j \\ &= \sum_{\ell \in \mathcal{I}} \mathcal{G}_\ell \left( \bigcirc_{j=1}^d U_j \mathbf{e}_{\ell_j} \right) \\ &= \sum_{\ell \in \mathcal{I}} \mathcal{G}_\ell \bigcirc_{j=1}^d (U_j)_{\ell_j} \end{aligned}$$

where we have used Lemma 1.3, and denoted the  $\ell_j$ -th column of  $U_j$  as  $(U_j)_{\ell_j}$ . We therefore have the sum of rank one tensors. There are  $\prod_{j=1}^d r_j$  possible values for  $\ell$  and so we have a CP decomposition of that rank. This provides an upper bound on CP rank, since the decomposition may not be optimal.  $\square$

On the other hand, given a Tucker decomposition of rank- $\mathbf{r}$ , one can instead regard this as a rank  $\prod_{i=1}^d r_i$  CP tensor, where each entry of the core is a weight to a rank-one tensor formed by the outer product of the respective columns of the factor matrices and then these are all summed up.

There is generically though no reason that these correspondences would be minimal in those respective decompositions. That is, given a minimal exact rank- $r$  CP decomposition, it does not follow that the Tucker decomposition used in Lemma 1.4 would be a minimal in terms of rank for the Tucker decomposition of that particular tensor.

Given an arbitrary tensor  $\mathcal{X} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$  of unknown Tucker rank it is often of interest to compute a good rank- $\mathbf{r}$  approximation of  $\mathcal{X}$ . In fact, an optimal Tucker rank- $\mathbf{r}$  minimizer  $[[\mathcal{X}]]_{\mathbf{r}}^{\text{opt}} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$

satisfying

$$\left\| \mathcal{X} - \llbracket \mathcal{X} \rrbracket_{\mathbf{r}}^{\text{opt}} \right\|_2 = \inf_{\substack{\mathcal{G} \in \mathbb{R}^{r_1 \times \dots \times r_d}, \\ U_j \in \mathbb{R}^{n_j \times r_j}}} \left\| \mathcal{X} - \llbracket \mathcal{G}, U_1, \dots, U_d \rrbracket \right\|_2 \quad (1.6)$$

always exists (see, e.g., Theorems 10.3 and Theorem 10.8 in [22]). However, computing such an  $\llbracket \mathcal{X} \rrbracket_{\mathbf{r}}^{\text{opt}}$  (which is not unique) is generally a challenging task that is accomplished only approximately via iterative techniques (see, e.g., [37]). As a result, in such situations one usually seeks to instead compute a *quasi-optimal* rank- $\mathbf{r}$  tensor  $\tilde{\mathcal{X}}$ .

**Definition 1.17.** The tensor  $\tilde{\mathcal{X}}$  of rank  $\mathbf{r}$  is quasi-optimal approximation of  $\mathcal{X}$  if it satisfies

$$\left\| \mathcal{X} - \tilde{\mathcal{X}} \right\|_2 \leq C \inf_{\substack{\mathcal{G} \in \mathbb{R}^{r_1 \times \dots \times r_d}, \\ U_j \in \mathbb{R}^{n_j \times r_j}}} \left\| \mathcal{X} - \llbracket \mathcal{G}, U_1, \dots, U_d \rrbracket \right\|_2, \quad (1.7)$$

where  $C \in \mathbb{R}^+$  is a positive constant independent of  $\mathcal{X}$ .

The following lemma demonstrates that one can recover a quasi-optimal rank- $\mathbf{r}$  approximation of an arbitrary tensor  $\mathcal{X} \in \mathbb{R}^{n_1 \times \dots \times n_d}$  by simply computing the left singular vectors of  $X_{[j]}$  for all  $j \in [d]$ . It is a variant of [22, Theorem 10.3]. We prove here for the sake of completeness.

**Lemma 1.5.** Let  $\mathcal{X} \in \mathbb{R}^{n_1 \times \dots \times n_d}$ , denote the  $i^{\text{th}}$ -largest singular value of  $X_{[j]}$  by  $\sigma_i(X_{[j]})$ , and define  $\Delta_{r,j} := \sum_{i=r+1}^{\tilde{n}_j} \sigma_i(X_{[j]})^2$  for all  $j \in [d]$  and  $r \in [\tilde{n}_j := \min\{n_j, \prod_{j \neq k} n_k\}]$ . Fix  $\mathbf{r} \in [\tilde{n}_1] \times \dots \times [\tilde{n}_d]$  and suppose that  $\llbracket \mathcal{X} \rrbracket_{\mathbf{r}}^{\text{opt}}$  satisfies (1.6). Then,

$$\sum_{j=1}^d \Delta_{r,j} \leq d \left\| \mathcal{X} - \llbracket \mathcal{X} \rrbracket_{\mathbf{r}}^{\text{opt}} \right\|_2^2.$$

*Proof.* Let  $P_j \in \mathbb{R}^{n_j \times n_j}$  be a rank  $r_j$  orthogonal projection matrix satisfying  $\|X_{[j]} - P_j X_{[j]}\|_F^2 = \Delta_{r,j} \leq \|X_{[j]} - Y\|_F^2$  for all rank at most  $r_j$  matrices  $Y$ .<sup>1</sup> Noting that  $Y = \left( \llbracket \mathcal{X} \rrbracket_{\mathbf{r}}^{\text{opt}} \right)_{[j]}$  will be a rank at most  $r_j$  matrix by (1.4), we then have that

$$\Delta_{r,j} \leq \left\| X_{[j]} - \left( \llbracket \mathcal{X} \rrbracket_{\mathbf{r}}^{\text{opt}} \right)_{[j]} \right\|_F^2 = \left\| \mathcal{X} - \llbracket \mathcal{X} \rrbracket_{\mathbf{r}}^{\text{opt}} \right\|_2^2 \quad \forall j \in [d].$$

Summing over the  $d$  values of  $j$  above now finishes the proof.  $\square$

Note, in order to simplify notation, we will often assume that  $n_j = n$  and  $r_j = r$  for all  $j \in [d]$ . That is our tensors will have equal side lengths and the rank will likewise be the same in every mode

<sup>1</sup>The Eckart-Young Theorem guarantees that we can compute such a  $P_j$  from the left singular values of  $X_{[j]}$ .

in the Tucker case, thus suppressing the need for additional level of sub-scripting. This will be done when modifying the proofs and results to deal with non-equal side lengths is straightforward; a bit of simplicity is achieved without too much damage to generality. Furthermore, we will principally be concerned with real valued tensors, rather than complex valued tensors.

#### 1.4.4 Randomized Numerical Linear Algebra Preliminaries

Concepts from randomized numerical linear algebra will feature prominently in our analysis of the embedding properties of different measurement operators.

**Definition 1.18** ( $(\epsilon, \delta, p)$ -Johnson-Lindenstrauss (JL) property). Let  $\epsilon > 0$ ,  $\delta \in (0, 1)$ , and  $p \in \mathbb{N}$ . A random matrix  $\Omega \in \mathbb{R}^{m \times n}$  has the  $(\epsilon, \delta, p)$ -Johnson-Lindenstrauss (JL) property<sup>2</sup> for an arbitrary set  $S \subset \mathbb{R}^n$  with cardinality at most  $p$  if it satisfies

$$(1 - \epsilon) \|\mathbf{x}\|_2^2 \leq \|\Omega \mathbf{x}\|_2^2 \leq (1 + \epsilon) \|\mathbf{x}\|_2^2 \quad \text{for all } \mathbf{x} \in S \quad (1.8)$$

with probability at least  $1 - \delta$ .

**Definition 1.19** (Set difference). Let  $\tilde{S} \subset \mathbb{C}^n$ , then the set difference of  $\tilde{S}$  denoted  $\tilde{S} - \tilde{S}$  is  $\{\mathbf{x} - \mathbf{y} | \mathbf{x}, \mathbf{y} \in \tilde{S}\} \in \mathbb{C}^n$

For all random matrix distributions discussed in this work, only an upper-bound on the cardinality of the set  $S$  appearing in Definition 1.8 is required in order to ensure with high probability that a given realization satisfies (1.8). No other property of the set  $S$  to be embedded will be required to define, generate, or apply  $\Omega$ . Such random matrices are referred to as *data oblivious*, or simply *oblivious*. The following variant of the famous Johnson-Lindenstrauss Lemma [32] demonstrates the existence of random matrices with the oblivious JL property. First though we can formally define a sub-Gaussian random variable and the sub-Gaussian norm, along with the related definition of Sub-exponential random variables.

**Definition 1.20** (Sub-exponential Random Variable). We say that  $X \in \mathbb{R}$  is sub-exponential random

---

<sup>2</sup>Formally, a distribution over  $m \times n$  matrices has the JL property if a matrix selected from this distribution satisfies (1.8) with probability at least  $1 - \delta$ . For brevity, here and in the next similar cases, the term “random matrix” will refer to a distribution over matrices.

variable if  $\exists \beta, \kappa > 0$  such that

$$P[|X| \geq t] \leq \beta e^{-\kappa t}, \forall t > 0$$

We can understand this as saying that the random variable decays exponentially.

**Definition 1.21** (Sub-Gaussian Random Variable). We say that  $X \in \mathbb{R}$  is sub-Gaussian random variable if  $\exists \beta, \kappa > 0$  such that

$$P[|X| \geq t] \leq \beta e^{-\kappa t^2}, \forall t > 0$$

There are many equivalences for a random variable being sub-Gaussian, see for example Proposition 2.5.2 in [63].

**Definition 1.22** (Sub-Gaussian norm). The sub-Gaussian norm of  $X$ , denoted  $\|X\|_{\psi_2}$  is the quantity

$$\|X\|_{\psi_2} := \inf_{t>0} \left\{ \mathbb{E}[\exp(X^2/t^2)] \leq 2 \right\}$$

There are many equivalences for a random variable being sub-Gaussian, most of which crucially are bounds which are related to the definition of the sub-Gaussian norm, see for example Proposition 2.5.2 in [63].

**Theorem 1.1** (Sub-Gaussian random matrices have the JL property). *Let  $S \subset \mathbb{R}^n$  be an arbitrary finite subset of  $\mathbb{R}^n$ . Let  $\delta, \epsilon \in (0, 1)$ . Finally, let  $\Omega \in \mathbb{R}^{m \times n}$  be a matrix with independent, mean zero, variance  $m^{-1}$ , sub-Gaussian entries all with sub-Gaussian norm less than or equal to  $c \in \mathbb{R}^+$ . Then*

$$(1 - \epsilon)\|\mathbf{x}\|_2^2 \leq \|\Omega\mathbf{x}\|_2^2 \leq (1 + \epsilon)\|\mathbf{x}\|_2^2$$

*will hold simultaneously for all  $\mathbf{x} \in S$  with probability at least  $1 - \delta$ , provided that*

$$m \geq \frac{C}{\epsilon^2} \ln \left( \frac{|S|}{\delta} \right), \tag{1.9}$$

*where  $C \leq 8c(16c + 1)$  is an absolute constant that only depends on the bound to the sub-Gaussian norm  $c$ .*

*Proof.* See, e.g., Lemma 9.35 in [18]. □

Next, we define a similar property for an infinite, yet rank constrained, set.

### 1.4.4.1 Oblivious Subspace Embedding (OSE) Results

In this section we will define the Oblivious Subspace Embedding property, and describe how to construct them from any random matrix distribution with the JL property.

**Definition 1.23.** [ $(\epsilon, \delta, r)$ -OSE property] Let  $\epsilon > 0$ ,  $\delta \in (0, 1)$ , and  $r \in \mathbb{N}$ . Fix an arbitrary rank  $r$  matrix  $A \in \mathbb{R}^{n \times N}$ . A random matrix  $\Omega \in \mathbb{R}^{m \times n}$  has the  $(\epsilon, \delta, r)$ -Oblivious Subspace Embedding (OSE) property for (the column space of)  $A$  if it satisfies

$$(1 - \epsilon) \|A\mathbf{x}\|_2^2 \leq \|\Omega A\mathbf{x}\|_2^2 \leq (1 + \epsilon) \|A\mathbf{x}\|_2^2 \quad \text{for all } \mathbf{x} \in \mathbb{R}^N \quad (1.10)$$

with probability at least  $1 - \delta$ .

Note that if  $Q \in \mathbb{R}^{n \times r}$  is an orthonormal basis for the column space of a rank  $r$  matrix  $A \in \mathbb{R}^{n \times N}$ , then the images of  $A$  and  $Q$  coincide, i.e.  $\{Q\mathbf{y} \mid \forall \mathbf{y} \in \mathbb{R}^r\} = \{A\mathbf{x} \mid \forall \mathbf{x} \in \mathbb{R}^N\}$ . Furthermore, since  $Q$  has orthonormal columns so that  $\|Q\mathbf{y}\|_2 = \|\mathbf{y}\|_2$  holds, (1.10) is equivalent to

$$(1 - \epsilon) \|\mathbf{y}\|_2^2 \leq \|\Omega Q\mathbf{y}\|_2^2 \leq (1 + \epsilon) \|\mathbf{y}\|_2^2 \quad \text{for all } \mathbf{y} \in \mathbb{R}^r \quad (1.11)$$

holding in this case. Below we will use the equivalence of (1.10) and (1.11) often by regularly constructing matrices satisfying (1.10) by instead constructing matrices satisfying (1.11).

The next result shows that random matrices with the JL property also have the OSE property. It is a standard result in the compressive sensing and randomized numerical linear algebra literature (see, e.g., [5, Lemma 5.1] or [57, Lemma 10]) that can be proven using an  $\epsilon$ -cover of the appropriate column space.

**Lemma 1.6** (Subspace embeddings from finite embeddings via a cover). *Fix  $\epsilon \in (0, 1)$ . Let  $\mathcal{L}_{\mathbf{y}}^r \subset \mathbb{R}^n$  be the  $r$ -dimensional subspace of  $\mathbb{R}^n$  spanned by an orthonormal basis  $\mathcal{Y}$ , and define  $S_{\mathbf{y}}^r := \left\{ \frac{\mathbf{x}}{\|\mathbf{x}\|_2} \mid \mathbf{x} \in \mathcal{L}_{\mathbf{y}}^r \setminus \{\mathbf{0}\} \right\}$ . Furthermore, let  $C \subset S_{\mathbf{y}}^r$  be a minimal  $(\frac{\epsilon}{16})$ -cover of  $S_{\mathbf{y}}^r \subset \mathcal{L}_{\mathbf{y}}^r$ . Then if  $\Omega \in \mathbb{C}^{m \times n}$  satisfies (1.8) with  $S \leftarrow C$  and  $\epsilon \leftarrow \frac{\epsilon}{2}$ , it will also satisfy*

$$(1 - \epsilon) \|\mathbf{x}\|_2^2 \leq \|\Omega \mathbf{x}\|_2^2 \leq (1 + \epsilon) \|\mathbf{x}\|_2^2 \quad \forall \mathbf{x} \in \mathcal{L}_{\mathbf{y}}^r. \quad (1.12)$$

*Proof.* See, e.g., Lemma 3 in [28] for this version. We discuss covering numbers in Section 1.4.4.3. □

Using Lemma 1.6 one can now easily prove the following corollary of Theorem 1.1 which demonstrates the existence of random matrices with the OSE property.

**Corollary 1.1.** *A random matrix  $\Omega \in \mathbb{R}^{m \times n}$  with mean zero, variance  $m^{-1}$ , independent sub-Gaussian entries has the  $(\epsilon, \delta, r)$ -OSE property for an arbitrary rank  $r$  matrix  $A \in \mathbb{R}^{n \times N}$  provided that*

$$m \geq C \frac{r}{\epsilon^2} \ln \left( \frac{C'}{\epsilon \delta} \right) \geq \frac{C}{\epsilon^2} \ln \left( \frac{\left( \frac{47}{\epsilon} \right)^r}{\delta} \right),$$

where  $C' > 0$  is an absolute constant, and  $C > 0$  is an absolute constant that only depends on the sub-Gaussian norms of  $\Omega$ 's entries.

*Proof.* As is done in Lemma 1.6, suppose  $S$  is a minimal  $\left(\frac{\epsilon}{16}\right)$ -cover of the  $r$  dimensional unit sphere in the column span of  $A$ . The cardinality of this cover is bounded by  $\left(\frac{47}{\epsilon}\right)^r$ . Apply Theorem 1.1 to the finite set  $S$ . □

The next Theorem and definition concern another important class of random, but highly structured, matrices that are known to satisfy the JL-embedding property.

**Theorem 1.2.** *Let  $U \in \mathbb{C}^{n \times n}$  be a unitary matrix with entries bounded by  $\max_{i,j \in [n]} |U_{i,j}| \leq \frac{K}{\sqrt{n}}$ . Let  $R \in \mathbb{C}^{m \times n}$  be a matrix obtained by selecting  $m$ -rows from the  $n \times n$  identity matrix i.i.d. uniformly at random and let  $D \in \mathbb{R}^{n \times n}$  be a diagonal matrix with i.i.d  $\pm 1$  Radamacher random values on its diagonal. Then*

$$\sqrt{\frac{n}{m}} R U D$$

will be an  $\epsilon$ -JL map of any given  $S \subset \mathbb{R}^n$  into  $\mathbb{C}^m$  with probability at least  $1 - \delta - n^{-\ln^3 n}$  provided that

$$m \geq c \frac{K^2}{\epsilon^2} \log \left( \frac{4|S|}{\delta} \right) \log^4 n$$

where  $c \in \mathbb{R}^+$  is an absolute constant.

Note that  $D$  can be applied in  $O(n)$  time to a vector in  $\mathbf{x} \in \mathbb{R}^n$ , since it involves reading the length of the vector and changing signs of some entries. The matrix  $R$  can be applied to a vector of length  $n$ ,  $UD\mathbf{x}$  in  $O(n)$ -time since it involves scanning the length of the vector and discarding values. Applying  $U$  then, since generically it should require at least reading in the inputs should be

at least  $\mathcal{O}(n)$ . Therefore the overall complexity is governed principally by  $U$ . If the unitary matrix  $U$  admits a fast matrix-vector, like for example using the Fast Fourier Transform (FFT) to effect a Discrete Fourier Transform, then  $U$  can be applied to  $D\mathbf{x}$  in  $\mathcal{O}(n \log n)$ -time.

In practice, the  $\log^4 n$  factor in the bound of  $m$  is often ignored with no change in performance.

The failure probability bound  $\delta + n^{-\ln^3 n}$  is a result of union bounding over two events –  $\frac{1}{\sqrt{m}}RF$  failing to have the yet undefined RIP property (see Definition 2.1) with probability at most  $n^{-\ln^3 n}$ ; the details of that can be found in Theorem 12.32 in [18]., and the probability of at most  $\delta$  that  $\sqrt{\frac{n}{m}}RFD$  fails to be an  $\frac{\epsilon}{2}$ -JL map; details of which can be found in Theorem 3.1 in [38]. Here, since we generally concern ourselves with  $n \gg 1$ , the failure probability can be made suitably small. We refer to this class of matrices as Subsampled Orthogonal Random Signs (SORS) matrices. The most frequently used type being as follows.

**Definition 1.24** (Subsampled Scrambled Fourier Transform (SSFT)). Let  $F$  be a unitary discrete Fourier transform matrix

$$F_{\ell,k} = \frac{1}{\sqrt{n}} e^{\frac{2\pi i \ell k}{n}}$$

If we take  $U = F$  and  $R$  and  $D$  be the matrices described in Theorem 1.2 then  $\sqrt{\frac{n}{m}}RFD$  is a JL map where

$$\max_{\ell,k} |F_{\ell,k}| = \frac{1}{\sqrt{n}}$$

so  $K = 1$ . The Fast Fourier Transform (FFT) can be used to apply  $F$  to any vector  $\mathbf{x} \in \mathbb{R}^n$  in  $\mathcal{O}(n \log n)$ .

This is the proto-typical Fast JL Matrix, though other choices for  $U$  are naturally possible. We also refer to it simply as  $RFD$  when context makes it clear.

Finally, the following lemma demonstrates that matrices satisfying (1.10) for  $A$  also approximately preserve the Frobenius norm of  $A$ . We present its proof here as an illustration of basic notation and techniques, and will use this result later in Chapter 3.

**Lemma 1.7.** *Suppose  $\Omega \in \mathbb{R}^{m \times n}$  satisfies (1.10) for a rank  $r$  matrix  $A \in \mathbb{R}^{n \times N}$ . Then,*

$$|\|A\|_F^2 - \|\Omega A\|_F^2| \leq \epsilon \|A\|_F^2.$$

*Proof.* Let  $\mathbf{e}_i \in \mathbb{R}^N$  denote the standard basis vector where the  $i$ -th entry is 1, and all others are zero (i.e.,  $\mathbf{e}_i$  is the  $i$ -th column of the  $N \times N$  identity matrix  $I_N$ ). Similarly let  $\mathbf{b}_i$  denote the  $i$ -th column of any given matrix  $B$ , and set  $\tilde{A} := \Omega A$ . By (1.10), we conclude that for all  $i \in [N]$  we have

$$\left| \|\mathbf{A}\mathbf{e}_i\|_2^2 - \|\Omega\mathbf{A}\mathbf{e}_i\|_2^2 \right| \leq \epsilon \|\mathbf{A}\mathbf{e}_i\|_2^2.$$

To establish the desired result, we represent the squared Frobenius norm of a matrix as the sum of the squared  $\ell_2$ -norms of its columns. Doing so, we see that

$$\begin{aligned} \left| \|A\|_F^2 - \|\Omega A\|_F^2 \right| &= \left| \sum_{i=1}^N \left( \|\mathbf{a}_i\|_2^2 - \|\tilde{\mathbf{a}}_i\|_2^2 \right) \right| \leq \sum_{i=1}^N \left| \|\mathbf{a}_i\|_2^2 - \|\tilde{\mathbf{a}}_i\|_2^2 \right| = \sum_{i=1}^N \left| \|\mathbf{A}\mathbf{e}_i\|_2^2 - \|(\Omega A)\mathbf{e}_i\|_2^2 \right| \\ &\leq \epsilon \sum_{i=1}^N \|\mathbf{A}\mathbf{e}_i\|_2^2 = \epsilon \|A\|_F^2. \end{aligned}$$

□

We will now define a property of random matrices related to fast approximate matrix multiplication.

#### 1.4.4.2 The Approximate Matrix Multiplication (AMM) Property

First proposed in [57, Lemma 6] (see also, e.g., [46] for other variants), the approximate matrix multiplication property will be crucial to our analysis of embedding properties of measurement operators.

**Definition 1.25** ( $(\epsilon, \delta)$ -AMM property). Let  $\epsilon > 0$  and  $\delta \in (0, 1)$ . A random matrix  $\Omega \in \mathbb{R}^{m \times n}$  satisfies the  $(\epsilon, \delta)$ -Approximate Matrix Multiplication property for two arbitrary matrices  $A \in \mathbb{R}^{p \times n}$  and  $B \in \mathbb{R}^{n \times q}$  if

$$\|A\Omega^T\Omega B - AB\|_F \leq \epsilon \|A\|_F \|B\|_F \quad (1.13)$$

holds with probability at least  $1 - \delta$ .

The following lemma can be used to construct random matrices with the AMM property from random matrices with the JL property. A slightly generalized version is proven in Appendix 3.5 for the sake of completeness.

**Lemma 1.8** (The JL property provides the AMM property). *Let  $A \in \mathbb{R}^{p \times n}$  and  $B \in \mathbb{R}^{n \times q}$ . There exists a finite set  $S \subset \mathbb{R}^n$  with cardinality  $|S| \leq 2(p + q)^2$  (determined entirely by  $A$  and  $B$ ) such*

that the following holds: If a random matrix  $\Omega \in \mathbb{R}^{m \times n}$  has the  $(\epsilon/2, \delta, 2(p+q)^2)$ -JL property for  $S$ , then  $\Omega$  will also have the  $(\epsilon, \delta)$ -AMM property for  $A$  and  $B$ .

*Proof.* Combine Lemma 3.12 with Remark 3.5. □

Using Lemma 1.8 one can now prove the following corollary of Theorem 1.1 which demonstrates the existence of random matrices with the AMM property for any two fixed matrices.

**Corollary 1.2.** Fix  $A \in \mathbb{R}^{p \times n}$  and  $B \in \mathbb{R}^{n \times q}$ . A random matrix  $\Omega \in \mathbb{R}^{m \times n}$  with mean zero, variance  $m^{-1}$ , independent sub-Gaussian entries will have the  $(\epsilon, \delta)$ -Approximate Matrix Multiplication property for  $A$  and  $B$  provided that

$$m \geq \frac{C}{\epsilon^2} \ln \left( \frac{2(p+q)^2}{\delta} \right),$$

where  $C > 0$  is an absolute constant that only depends on the sub-Gaussian norms/parameters of  $\Omega$ 's entries.

*Proof.* Apply Theorem 1.1 to the finite set  $S$  guaranteed by Lemma 1.8. □

### 1.4.4.3 Covering Numbers

We state some results concerning covering numbers, which are useful for showing  $\epsilon$ -JL maps apply to more general infinite sets, and are a main ingredient to several of the results referred to above, such as Lemma 1.6. We begin then with these definitions and then show how to obtain covering number bounds for subsets, which is a main ingredient in net arguments used for embedding infinite sets.

**Definition 1.26** ( $\delta$ -cover). Let  $X$  be a normed metric space over the complex numbers and  $T \subseteq \mathbb{C}^N$ . A  $\delta$ -cover of  $T$  with respect to norm  $\|\cdot\|_X$  is a subset of  $S \subseteq T$  such that  $\forall \mathbf{x} \in T, \exists \mathbf{y} \in S$  with  $\|\mathbf{x} - \mathbf{y}\|_X < \delta$  where

$$T \subseteq \bigcup_{\mathbf{y} \in S} B_X(\mathbf{y}, \delta)$$

Note that  $B_X(\mathbf{y}, \delta)$  is the open ball with center  $\mathbf{y} \in \mathbb{C}^N$  and radius  $\delta$  with respect to the norm  $\|\cdot\|_X$ . Usually it will be clear from context the space and norm, and so we'll simplify notation and write instead  $B(\mathbf{y}, \delta)$

**Definition 1.27** ( $\delta$ -covering Number). The  $\delta$ -covering number of  $T \subseteq \mathbb{C}^N$ , denoted  $C_\delta^X(T)$  with respect to  $\|\cdot\|_X$  is the smallest integer such that there exists a  $\delta$ -cover  $S \subseteq T$  where  $|S| = C_\delta^X(T)$ .

If no such integer exists we say that  $C_\delta^X(T) = \infty$

**Definition 1.28** ( $\delta$ -packing). Let  $T \subseteq \mathbb{C}^N$ . A  $\delta$ -packing of  $T$  with respect to norm  $\|\cdot\|_X$  is a subset of  $S \subseteq T$  such that  $\forall \mathbf{x}, \mathbf{y} \in S, \mathbf{x} \neq \mathbf{y}$  with  $\|\mathbf{x} - \mathbf{y}\|_X \geq \delta$  then

$$B_X(\mathbf{x}, \delta/2) \cap B_X(\mathbf{y}, \delta/2) = \emptyset$$

**Definition 1.29** ( $\delta$ -packing Number). The  $\delta$ -packing number of  $T \subseteq \mathbb{C}^N$ , denoted  $P_\delta^X(T)$  with respect to  $\|\cdot\|_X$  is the largest integer such that there exists a  $\delta$ -packing  $S \subseteq T$  where  $|S| = P_\delta^X(T)$ .

If no such integer exists we say that  $P_\delta^X(T) = \infty$

**Lemma 1.9.** Let  $T \subseteq \mathbb{C}^N$  and  $\delta \in (0, \infty)$ . Then

$$P_{2\delta}^X(T) \leq C_\delta^X(T) \leq P_\delta^X(T)$$

*Proof.* Let  $P_{2\delta} \subset T$  be a maximal  $2\delta$ -packing of  $T$  and  $C_\delta \subseteq T$  be a minimal  $\delta$ -cover of  $T$ . Each point  $\mathbf{x} \in P_{2\delta}$  is closest to a different point  $\mathbf{y} \in C_\delta$ . To see this, suppose to the contrary that for  $\mathbf{x}_1, \mathbf{x}_2 \in P_{2\delta}, \mathbf{x}_1 \neq \mathbf{x}_2$  there was some point  $\mathbf{y} \in C_\delta$  such that  $\mathbf{x}_1, \mathbf{x}_2 \in B(\mathbf{y}, \delta)$  this implies that  $\mathbf{y} \in B_X(\mathbf{x}_1, \delta) \cap B_X(\mathbf{x}_2, \delta)$  which is a contradiction.

So, since each point in  $P_\delta$  can be identified with at least one point in  $C_\delta$ . We thus define an injection  $f : P_{2\delta} \rightarrow C_\delta$  where  $f(\mathbf{x}) = \mathbf{y}, \mathbf{x} \in B(\mathbf{y}, \delta)$ . Since  $f$  is an injection, we have that the cardinality of  $C_\delta$  must be equal to or larger than  $P_{2\delta}$ , which is equivalent to the left hand side of the desired inequality.

Next, suppose  $P_\delta$  is a maximal  $\delta$ -packing of  $T$ . Now suppose for eventual contradiction that there exists a point  $\mathbf{y} \in T, \mathbf{y} \notin P_\delta$  such that  $\|\mathbf{x} - \mathbf{y}\| \geq \delta, \forall \mathbf{x} \in P_\delta$ . This implies that  $B_X(\mathbf{x}, \delta/2) \cap B_X(\mathbf{y}, \delta/2) = \emptyset$ . Thus  $P_\delta \cup \{\mathbf{y}\}$  is a  $\delta$ -packing of  $T$ . This contradicts that  $P_\delta$  is maximal. So, for all points  $\mathbf{y} \in T$ , there is  $\mathbf{x} \in P_\delta$  such that  $\|\mathbf{x} - \mathbf{y}\| \leq \delta$ , which is to say  $P_\delta$  is a  $\delta$ -covering of  $T$ , and therefore the cardinality of  $P_\delta$  is equal to or larger than the  $\delta$ -covering number for  $T$ . This is the right hand side of the desired inequality.  $\square$

**Lemma 1.10.** Let  $T \subseteq \mathbb{R}^N$  and  $\delta \in (0, \infty)$ . Furthermore let  $B$  denote the unit ball  $B_X(0, 1)$  in  $\mathbb{R}^N$  with respect to some norm  $\|\cdot\|_X$ . Then

$$\left(\frac{1}{\delta}\right)^N \frac{\text{Vol}(T)}{\text{Vol}(B)} \leq C_\delta^X(T) \leq P_\delta^X(T) \leq \left(\frac{2}{\delta}\right)^N \frac{\text{Vol}(T + \left(\frac{\delta}{2}\right) B)}{\text{Vol}(B)}$$

holds, where  $\text{Vol}(T) = \int_T 1 dV$ , the Lebesgue measure of  $T$  in  $\mathbb{R}^N$ .

Note that addition of sets is syntactical sugar for set difference of certain sets, i.e.  $T + S = T - (-S) = \{t + s \mid \forall t \in T, s \in S\}$

*Proof.* Suppose  $C_\delta$  is a minimal  $\delta$  cover of  $T$ . By definition then of  $\delta$ -cover

$$T \subseteq \bigcup_{\mathbf{y} \in C_\delta} B(\mathbf{y}, \delta)$$

So, using sub-additivity of measurable sets, translation invariance, and scaling we have

$$\text{Vol}(T) \leq \text{Vol}\left(\bigcup_{\mathbf{y} \in C_\delta} B(\mathbf{y}, \delta)\right) \leq C_\delta^X \text{Vol}(B(\mathbf{y}, \delta)) = C_\delta^X \delta^N \text{Vol}(B(0, 1))$$

Rearranging terms, we obtain the left hand side of the desired inequality

$$\left(\frac{1}{\delta}\right)^N \frac{\text{Vol}(T)}{\text{Vol}(B)} \leq C_\delta^X(T)$$

Now suppose  $P_\delta$  is a maximal  $\delta$ -packing of  $T$ . It follows that

$$\bigcup_{\mathbf{y} \in P_\delta} B(\mathbf{y}, \delta/2) \subset T + B(0, \delta/2)$$

Since the balls that make up the  $\delta$ -packing of  $T$  are disjoint, we have that their measure is additive.

Again, using translation invariance and scaling, this implies

$$\text{Vol}\left(\bigcup_{\mathbf{y} \in P_\delta} B(\mathbf{y}, \delta/2)\right) = P_\delta^X(T) \left(\frac{\delta}{2}\right)^N \text{Vol}(B(\mathbf{0}, 1)) \leq \text{Vol}(T + B(0, \delta/2))$$

Which after rearranging terms matches the right hand side of the desired inequality □

**Corollary 1.3.**  $\left(\frac{1}{\delta}\right)^N \leq C_\delta^X(B) \leq \left(1 + \frac{2}{\delta}\right)^N$  for all norms  $\|\cdot\|_X$  on  $\mathbb{R}^N$

*Proof.* We can apply lemma 1.10 where  $T = B(0, 1)$ . So,

$$\left(\frac{1}{\delta}\right)^N \frac{\text{Vol}(B)}{\text{Vol}(B)} \leq C_\delta^X(T)$$

yields the first half of the corollary. Next, note that using  $T = B(0, 1)$  we see that

$$B(0, 1) + B\left(0, \frac{\delta}{2}\right) \subseteq B\left(0, \left(1 + \frac{\delta}{2}\right)\right)$$

Note that by scaling, we have the following for the volume calculation

$$\text{Vol}\left(B\left(0, \left(1 + \frac{\delta}{2}\right)\right)\right) = \left(1 + \frac{\delta}{2}\right)^N \text{Vol}(B(0, 1))$$

Putting this into the previous lemma, we see

$$C_\delta^X(T) \leq \left(\frac{2}{\delta}\right)^N \left(1 + \frac{\delta}{2}\right)^N \frac{\text{Vol}(B(0, 1))}{\text{Vol}(B)} = \left(1 + \frac{2}{\delta}\right)^N$$

Which completes the second inequality □

Note that if we add the assumption in the corollary that  $\delta \in (0, 1)$  then we can bound  $\left(1 + \frac{2}{\delta}\right) \leq \left(\frac{1}{\delta} + \frac{2}{\delta}\right) = \frac{3}{\delta}$ , for a more concise, though less tight bound.

**Corollary 1.4.** *If  $S \subseteq \overline{B_X(0, 1)} \subset \mathbb{R}^N$  then*

$$C_\delta^X(T) \leq \left(1 + \frac{2}{\delta}\right)^N$$

*Proof.* By observing

$$\text{Vol}\left(S + B\left(0, \frac{\delta}{2}\right)\right) \leq \text{Vol}\left(\overline{B_X(0, 1)} + B\left(0, \frac{\delta}{2}\right)\right)$$

and applying the same reasoning as Corollary 1.3, we achieve the result. □

## 1.5 Modewise Measurements Framework

For Chapters 2 and 3, we will be investigating specializations of the following measurement framework for tensor recovery. The overall measurement process we will consider is a collection of one or more measurement procedures that interleaves reshaping of the tensor (see Definition 1.7) with modewise measurements using modewise products with linear maps that are applied to some number of (reshaped) modes (see Definition 1.10). Naturally, it is formally possible to layer as many rounds of reshaping and measurement as one wishes, however we will be concerned with measurement procedures that have at most two stages of reshaping and linear transforms applied the fibers of the reshaped tensor.

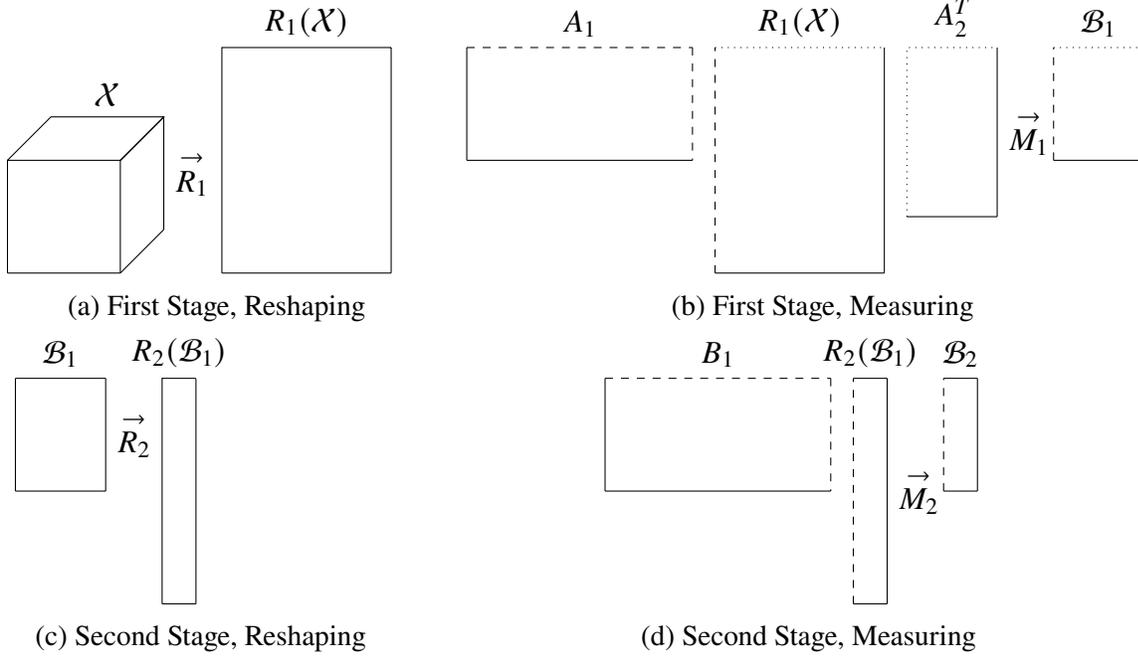


Figure 1.3 Schematic of a two-stage measurement procedure. The first reshaping  $R_1$  takes a three mode tensor and reshapes it into a two mode tensor. The modewise measurements then in the measurement phase of the first stage corresponds to left and right multiplication by the measurement maps  $A_1$  and  $A_2^T$  to form intermediary tensor  $\mathcal{B}_1$ . The second stage depicts a vectorization reshaping operation  $R_2$  and a final measurement phase where the matrix  $B_1$  is applied to the remaining mode to form the final set of measurements,  $\mathcal{B}_2$ .

For example, a generic two-stage modewise linear operator  $\mathcal{L} : \mathbb{R}^{n_1 \times \dots \times n_d} \rightarrow \mathbb{R}^{m_1 \times \dots \times m_{d'}}$  could take the form

$$\mathcal{L}(\mathcal{X}) := R_2 \left( R_1(\mathcal{X}) \times_1 A_1 \cdots \times_{\tilde{d}} A_{\tilde{d}} \right) \times_1 B_1 \cdots \times_{d'} B_{d'}, \quad (1.14)$$

where  $R_1$  is a reshaping operator which rearranges elements of  $\mathbb{R}^{n_1 \times \dots \times n_d}$  tensor into an  $\mathbb{R}^{\tilde{n}_1 \times \dots \times \tilde{n}_{\tilde{d}}}$ . After this reshaping, using modewise product (Definition 1.10),  $A_j \in \mathbb{R}^{\tilde{m}_j \times \tilde{n}_j}$  is applied to the reshaped tensor for  $j \in [\tilde{d}]$ . This is followed by an additional reshaping via  $R_2$  into an  $\mathbb{R}^{m'_1 \times \dots \times m'_{d'}}$  tensor, and final set of  $j$ -mode products using the matrices  $B_j \in \mathbb{R}^{m_j \times m'_j}$  for  $j \in [d']$ . See Figure 1.3 for a schematic depiction of a two-stage operator of this type.

Clearly, any  $q$ -stage modewise operators can be defined in this way. This view of measurement operators was first analyzed in [28, 30]. Immediately, we can see multi-staged modewise compression operators where several modes are compressed can offer a computational advantages

over standard vector-based approaches, provided the measurements are sufficiently sized to retain information about the data. By vector-based approaches, what we mean is the direct application of classic compressive sensing results which have been applied to vectors; whereby one simply takes the tensor data and as a first step, rearranges its elements into a vector. This approach fits the framework, where  $R_1$  is a vectorization operator as in Definition 1.5. That is,  $\tilde{d} = 1$ ,  $A_1 = A \in \mathbb{R}^{m \times \prod_j^d n_j}$  is a Johnson-Lindenstrauss map for example, and all remaining operators  $\mathcal{R}_2, B_1$ , etc. are the identity (see, e.g., [49]).

If instead  $R_1$  is a reshaping operation that combines even a few, but not all modes, then the resulting modewise linear transforms can be formed using significantly fewer random variables (effectively, independent random bits), and stored using less memory by avoiding the use of a single, comparatively large  $m \times \prod_j^d n_j$  measurement matrix as in the vectorized-first approach. Indeed as we will discuss in Chapter 2, simply forming or storing such a measurement matrix can be impractical, whereas one can achieve accurate recovery by instead using several, smaller, modewise maps.

In addition to the storage aspect of the map themselves, modewise linear operators of this kind are also straightforward to parallelize and offer the possibility of choosing different maps at a more granular level to better respect the multi-modal structure of the given tensor data.

## CHAPTER 2

### TENSOR RESTRICTED ISOMETRY PROPERTY AND TENSOR ITERATIVE HARD THRESHOLDING FOR RECOVERY

We are now equipped to describe how we applied the measurement framework described in Section 1.5 with a recovery procedure in order to produce a method that provably can yield reliable and accurate results for the tensor recovery problem. The concept that will be of central importance to this endeavor is the Tensor Restricted Isometry Property (TRIP). To explain where this term comes from and why it is associated with recovery of an object from measurements, we will recall the term from which it originates, namely the Restricted Isometry Property (RIP). Classically RIP is nothing more than norm preserving Johnson-Lindenstrauss property applied to the set of sparse vectors. That is, we have the definition,

**Definition 2.1.** [RIP( $\epsilon, S$ ) property] We say that a linear map  $\mathcal{A}$ , defined on a normed vector space with norm  $\|\cdot\|$ , has the RIP( $\epsilon, S$ ) property if for all elements  $s \in S$

$$(1 - \epsilon)\|s\|^2 \leq \|\mathcal{A}(s)\|^2 \leq (1 + \epsilon)\|s\|^2 \tag{2.1}$$

As we mentioned, classically in compressive sensing, the set under consideration are the  $s$ -sparse vectors; and so  $S = K_s$ , where

$$K_s = \{\mathbf{x} \in \mathbb{C}^N \mid \|\mathbf{x}\|_0 \leq s\} = \bigcup_{S \subseteq [N] \mid |S| \leq s} \text{span} \{\mathbf{e}_j\}_{j \in S}$$

though we emphasize the definition here admits any subset from a normed vector space. Famously however in the case of  $K_s$ ,  $\ell_1$ -minimization by basis pursuit was shown to be able to recover the original signal vector (exactly or approximately depending on the situation) from a small number of measurements, when the measurement operator satisfies this approximate geometry preserving property and moreover that common distributions of random matrices provably had RIP with high probability, e.g. see Chapter 4 of [18]. There are other well known methods for the recovery part of the measure and recover compressive sensing problem, such as orthogonal matching pursuit and iterative hard thresholding (IHT). Extending these results to low rank matrices was a natural direction for this idea, and was the subject of celebrated research in the field of compressive

sensing, see [14] for an overview. In this chapter however we shall be concerned with the extension of low-rank recovery to higher-order tensors.

## 2.1 Two-Stage Modewise Measurements and Tensor Restricted Isometry Property

Prior research, such as [21, 20], has extended the Tensor IHT method (TIHT) to low-rank CP and Tucker tensors. TIHT is an iterative method that consists alternating updates where the first sub-step applies the adjoint of the measurement operator to the residual of the last iterate and a second sub-step that thresholds that update back to the relevant constraint set; the set of low-rank tensors. That is, we take a gradient step first, and then project back onto the proper constraint set for our problem. This is very recognizable as a direct extension of IHT to the case of higher-order tensors. That is, we have the update rule

$$\begin{aligned}\mathcal{Y}^j &= \mathcal{X}^j + \mu_j \mathcal{L}^*(\mathcal{L}(\mathcal{X}) - \mathcal{L}(\mathcal{X}^j)), \\ \mathcal{X}^{j+1} &= \mathcal{H}_{\mathbf{r}}(\mathcal{Y}^j).\end{aligned}\tag{2.2}$$

where  $j$  denotes the iteration,  $\mathcal{L}$  and  $\mathcal{L}^*$  denote the measurement operator and its adjoint,  $\mathcal{X}^0$  is the initialization (randomly generated for example), the symbol  $\mathcal{H}_{\mathbf{r}}$  is the thresholding operator, that takes the tensor and finds a Tucker decomposition of the specified rank that approximates the iterate  $\mathcal{Y}^j$ , and finally  $\mu_j$  is a step-size parameter. The optimality of the thresholding operation is something we will have cause to comment on later in this chapter.

The following theorem is one of the main results of [56]. It implies accurate reconstruction of a tensor  $\mathcal{X}$  is possible via TIHT when the measurement operator satisfies the yet to be defined  $\text{TRIP}(\delta, 3\mathbf{r})$  property for a sufficiently small  $\delta$  and a sufficient number of iterations, with a reasonable initialization since the update rule is contractive.

**Theorem 2.1** ([56], Theorem 1). *Let  $\mathcal{X} \in \mathbb{R}^{n_1 \times \dots \times n_d}$ , let  $0 < a < 1$ , and let  $\mathcal{L} : \mathbb{R}^{n_1 \times \dots \times n_d} \rightarrow \mathbb{R}^m$  satisfy  $\text{TRIP}(\delta, 3\mathbf{r})$  with  $\delta < a/4$  for some  $a \in (0, 1)$ . Assume that  $\mathbf{y} = \mathcal{L}(\mathcal{X}) + \mathbf{e}$ , where  $\mathbf{e} \in \mathbb{R}^m$  is an arbitrary noise vector, and let  $\mathcal{X}^j$  and  $\mathcal{Y}^j$  be defined as in (2.2). Assume that*

$$\|\mathcal{Y}^j - \mathcal{X}^{j+1}\|_2 \leq (1 + \xi_a) \|\mathcal{Y}^j - \mathcal{X}\|_2, \quad \text{where} \quad \xi_a = \frac{a^2}{17(1 + \sqrt{1 + \delta}) \|\mathcal{L}\|_{2 \rightarrow 2}}.\tag{2.3}$$

Then

$$\|\mathcal{X}^{j+1} - \mathcal{X}\|_2 \leq a^j \|\mathcal{X}^0 - \mathcal{X}\|_2 + \frac{b_a}{1-a} \|\mathbf{e}\|_2,$$

where  $b_a = 2\sqrt{1+\delta} + \sqrt{4\xi_a + 2\xi_a^2} \|\mathcal{L}\|_{2 \rightarrow 2}$ .

The assumption (2.3) in the Theorem 2.1 is about the quality of the thresholding operation in (2.2). This assumption is unfortunately stronger than the quasi-optimality guarantee which say truncated Higher Order SVD computed via Algorithm 1.1 is able to satisfy, as shown in Lemma 1.5. However, the Lemma is an upper-bound, and in practice we have observed that the truncated HOSVD procedure does produce estimates that are very good in some cases; though naturally in real world situations it would be impossible to assess quantitatively that without finding an optimal HOSVD - which is difficult. Some other choice for thresholding operation, to include refinements using iterations of HOOI or some other method can likewise be employed in practice to shore up one's confidence that the requirement is met. Practically speaking, it is to some degree a matter of how much computational effort is reasonable to budget for the task.

We note that in [56], the authors show it is possible to draw maps from random distributions which will satisfy the desired TRIP property with high probability. Unfortunately, these maps require first vectorizing the input tensor into a  $n^d$ -dimensional vector and then multiplying by an  $m \times n^d$  matrix as we have mentioned earlier. Such a map requires  $m$ -times more memory than the original tensor and thus can be unsuitable in the big data scenario; and indeed as we shall discuss in our experiments, we were able to recover large tensors at better accuracy using a modewise approach where a vectorized approach failed given the same constraints on memory.

With Theorem 2.1 in mind, we will now can move to specify the class of measurement operators  $\mathcal{L}$  we wish to use in this chapter. Two additional preliminaries await. First, the elaboration now on the RIP definition alluded to, so that it includes higher order tensors with a low-rank decomposition.

**Definition 2.2.** [TRIP( $\delta, \mathbf{r}$ ) property] We say that a linear map  $\mathcal{A}$  has the TRIP( $\delta, \mathbf{r}$ ) property if for all  $\mathcal{X}$  with HOSVD rank at most  $\mathbf{r}$  we have

$$(1 - \delta) \|\mathcal{X}\|_2^2 \leq \|\mathcal{A}(\mathcal{X})\|_2^2 \leq (1 + \delta) \|\mathcal{X}\|_2^2 \tag{2.4}$$

Our second remaining piece is to establish how the HOSVD of a tensor is related to the HOSVD of its reshaping. Let  $\kappa$  be an integer which divides  $d$ , the number of modes, and let  $d' := d/\kappa$ . Consider the reshaping operator

$$\mathcal{R}_1 : \bigotimes_{i=1}^d \mathbb{R}^{n_i} \rightarrow \bigotimes_{i=1}^{d'} \mathbb{R}^{n_{\kappa i}}$$

that combines  $\kappa$  sized sets of consecutive modes of a tensor into one mode, (see Definition 1.7). For example if the tensor has four modes, and  $\kappa = 2$  then the reshaping will pair the first and second modes together and the third and fourth together and thus reshape the four mode tensor into a matrix of size  $n_1 n_2 \times n_3 n_4$ . Any procedure to group the modes would be valid; so there is no loss in generality for assuming consecutive modes being grouped. Formally,  $\mathcal{R}_1$  is defined to be the unique linear operator such that on rank-one tensors it acts as

$$\mathcal{R}_1 \left( \bigcirc_{i=1}^d \mathbf{x}^{(i)} \right) := \bigcirc_{i=1}^{d'} \left( \bigotimes_{\ell=1+\kappa(i-1)}^{\kappa i} \mathbf{x}^{(\ell)} \right) =: \bigcirc_{i=1}^{d'} \overset{\circ}{\mathbf{x}}^{(i)},$$

recall  $\otimes$  denotes the Kronecker product, see Definition 1.11. We observe that if a tensor  $\mathcal{X}$  has a Tucker decomposition (see (1.3)), then its reshaping  $\overset{\circ}{\mathcal{X}} := \mathcal{R}_1(\mathcal{X})$  is the  $d'$ -mode tensor  $\overset{\circ}{\mathcal{X}} \in \bigotimes_{i=1}^{d'} \mathbb{R}^{n_{\kappa i}}$  with HOSVD rank at most  $\mathbf{r}' := (r^\kappa, \dots, r^\kappa)$  given by

$$\overset{\circ}{\mathcal{X}} = \sum_{j_{d'}=1}^{r^\kappa} \dots \sum_{j_1=1}^{r^\kappa} \overset{\circ}{\mathcal{G}}(j_1, \dots, j_{d'}) \bigcirc_{\ell=1}^{d'} \overset{\circ}{\mathbf{u}}_{j_\ell}^{(\ell)}, \quad (2.5)$$

where the new component vectors  $\overset{\circ}{\mathbf{u}}_{j_\ell}^{(\ell)}$  are obtained by taking Kronecker product of the appropriate  $\mathbf{u}_{k_i}^{(i)}$ , and where  $\overset{\circ}{\mathcal{G}} \in \mathbb{R}^{r^\kappa \times \dots \times r^\kappa}$  is a reshaped version of  $\mathcal{G}$  from (1.3).

Since each of the  $\{\mathbf{u}_{k_i}^{(i)}\}_{k_i=1}^{r^\kappa}$  was an orthonormal basis for the space spanned by  $U_i$ , it follows that  $\{\overset{\circ}{\mathbf{u}}_{j_i}^{(i)}\}_{j_i=1}^{r^\kappa}$  is an orthonormal basis for the space  $\text{span} \left( \{\overset{\circ}{\mathbf{u}}_{j_i}^{(i)}\}_{j_i=1}^{r^\kappa} \right)$ . That is, the Kronecker product of orthonormal vectors is itself an orthonormal vector in its respective vector space.

Let  $A_i$  be an  $m \times n^\kappa$  matrix for  $i \in [d']$ , denote  $\mathcal{A} : \mathbb{R}^{n^\kappa \times \dots \times n^\kappa} \rightarrow \mathbb{R}^{m \times \dots \times m}$  to be the linear map which acts modewise on  $d'$ -mode tensors by way of the modewise products:

$$\mathcal{A}(\mathcal{Y}) = \mathcal{Y} \times_1 A_1 \times_2 \dots \times_{d'} A_{d'}. \quad (2.6)$$

Let  $\mathcal{X}$  be a  $d$ -mode tensor with HOSVD decomposition given by (1.16). By Lemma 1.1 and (2.5), we have that

$$\mathcal{A}(\mathcal{R}_1(\mathcal{X})) = \mathcal{A}(\overset{\circ}{\mathcal{X}}) = \sum_{j_{d'}=1}^{r^\kappa} \dots \sum_{j_1=1}^{r^\kappa} \overset{\circ}{G}(j_1, \dots, j_{d'}) (A_1 \overset{\circ}{\mathbf{u}}_{j_1}^{(1)} \circ \dots \circ A_{d'} \overset{\circ}{\mathbf{u}}_{j_{d'}}^{(d')}). \quad (2.7)$$

Next, we define the second stage of reshaping and modewise operations, similar to what was discussed in the example in Section 1.5. Let  $\mathcal{R}_2$  denote the reshaping operation:

$$\mathcal{R}_2 : \bigotimes_{i=1}^{d'} \mathbb{R}^m \rightarrow \mathbb{R}^{m^{d'}}, \quad (2.8)$$

that is the reshaping operation which vectorizes the tensor. Next, Let  $B$  be a  $m_{2nd} \times m^{d'}$  linear map. Denote

$$\mathcal{A}_{2nd}(\mathcal{X}) := B \cdot \mathcal{R}_2(\mathcal{A}(\mathcal{R}_1(\mathcal{X}))), \quad (2.9)$$

The main theoretical result that the chapter relies on is to show that  $\mathcal{A}_{2nd}$  satisfies the TRIP property. This relies on showing how  $A_i$  satisfies a restricted isometry property on the set  $S_{1,2}$ , to be defined below, and how the gestalt of the measurement operator inherits its properties from its component parts.

We will require two more definitions.

**Definition 2.3.** [The set  $S_{1,2}$ ] Consider a set of vectors in  $\mathbb{R}^{n^\kappa}$ ,

$$S_1 := \{\overset{\circ}{\mathbf{u}} \mid \overset{\circ}{\mathbf{u}} = \otimes_1^\kappa \mathbf{u}^{(i)}, \mathbf{u}^{(i)} \in \mathbb{S}^{n-1}\}, \quad (2.10)$$

and let  $S_2 := \left\{ \frac{\mathbf{x}+\mathbf{y}}{\|\mathbf{x}+\mathbf{y}\|_2} \mid \mathbf{x}, \mathbf{y} \in S_1 \text{ s.t. } \langle \mathbf{x}, \mathbf{y} \rangle = 0 \right\}$ . Denote  $S_{1,2} := S_1 \cup S_2$ , and observe that  $S_{1,2} \subseteq \mathbb{S}^{n^\kappa-1}$ .

**Definition 2.4** (Nearly orthogonal tensors  $\mathcal{B}_{R,\mu,\theta,r}$ ). Let  $\mathcal{B}_{R,\mu,\theta,r}$  be the set of  $d$ -mode tensors in  $\mathcal{X} \in \mathbb{R}^{n \times \dots \times n}$  that may be written in Tucker form (1.15) such that

- (a)  $\|\mathbf{u}_{k_i}^{(i)}\|_2 \leq R$  for all  $i$  and  $k_i$ ,
- (b)  $|\langle \mathbf{u}_{k_i}^{(i)}, \mathbf{u}_{k'_i}^{(i)} \rangle| \leq \mu$  for all  $k_i \neq k'_i$ ,
- (c) the core tensor  $\mathcal{G}$  satisfies  $\|\mathcal{G}\|_2 = 1$ ,
- (d)  $\mathcal{G}$  has orthogonal subtensors in the sense that (1.5) holds for all  $1 \leq i \leq d$ ,

(e)  $\|\mathcal{X}\|_2 \geq \theta$ .

**Theorem 2.2.** *Let  $r \geq 2$  and let  $\mathbf{r} = (r, r, \dots, r) \in \mathbb{R}^d$ . Suppose that  $\mathcal{A}$  and  $\mathcal{A}_{2nd}$  are defined as in (2.6) and (2.9). Let  $d' = d/\kappa$  and assume that  $A_i$  satisfies the  $\text{RIP}(\epsilon, S_{1,2})$  property for all  $i = 1, \dots, d'$ , where  $\delta = 12d'r^d\epsilon < 1$ . Assume that  $A_{2nd}$  satisfies the  $\text{RIP}(\delta/3, \mathcal{B}_{1+\epsilon, \epsilon, 1-\delta/3, \mathbf{r}'})$  property where  $\mathbf{r}' = (r^\kappa, \dots, r^\kappa) \in \mathbb{R}^{d'}$ . Then,  $\mathcal{A}_{2nd}$  will satisfy the  $\text{TRIP}(\delta, \mathbf{r})$  property, i.e.,*

$$(1 - \delta)\|\mathcal{X}\|_2^2 \leq \|\mathcal{A}_{2nd}(\mathcal{X})\|_2^2 \leq (1 + \delta)\|\mathcal{X}\|_2^2$$

for all  $\mathcal{X}$  with HOSVD rank at most  $\mathbf{r}$ .

The proof of this result depends on several intermediary results, so we delay its presentation to Section 2.4 of this chapter. We instead in this section wish to focus on the following corollary, and a discussion with critiques based on empirical findings and several observations we made in the course of its investigation.

**Corollary 2.1.** *Assume the operator  $\mathcal{L}$ , is defined in one of the following ways:*

- (a)  $\mathcal{L} = \text{vec} \circ \mathcal{A} \circ \mathcal{R}$ , where  $\mathcal{A}$  is defined as per (2.6), and the matrices  $A_i$  satisfy the  $\text{RIP}(\epsilon, S_{1,2})$ , and  $\delta = 4d'(3r)^d\epsilon < a/4$ .
- (b)  $\mathcal{L} = \mathcal{A}_{2nd}$  defined as in (2.9), its component matrices  $A_i$  satisfy  $\text{RIP}(\epsilon, S_{1,2})$  property,  $\delta = 12d'^2(3r)^d\epsilon$ , and  $A_{\text{final}}$  satisfies the  $\text{RIP}(\delta/3, \mathcal{B}_{1+\epsilon, \epsilon, 1-\delta/3, \mathbf{r}'})$  property.

Consider the recovery problem from the noisy measurements  $\mathbf{y} = \mathcal{L}(\mathcal{X}) + \mathbf{e}$ , where  $\mathbf{e} \in \mathbb{R}^m$  is an arbitrary noise vector. Let  $0 < a < 1$ , and let  $\mathcal{X}^j$ , and  $\mathcal{Y}^j$  be defined as in (2.2), and assume that (2.3) holds. Then,

$$\|\mathcal{X}^{j+1} - \mathcal{X}\|_2 \leq a^j \|\mathcal{X}^0 - \mathcal{X}\|_2 + \frac{b_a}{1-a} \|\mathbf{e}\|_2,$$

where  $b_a = 2\sqrt{1+\delta} + \sqrt{4\xi_a + 2\xi_a^2} \|\mathcal{A}\|_{2 \rightarrow 2}$ .

Corollary 2.1 encapsulates the theoretical basis for why we can expect a two stage modewise measurement operator like  $\mathcal{A}_{2nd}$ , where the constituent measurement maps satisfy the RIP properties, will yield a contractive iterative method in the presence of some noise. A particular choice of measurement map, such as i.i.d sub-Gaussian matrices then produces row bounds - provided we can bound the size of covers of sets  $S_{1,2}$  and  $\mathcal{B}_{R, \mu, \theta, \mathbf{r}}$ . The intermediary results that describe these

details for how the properties of the various parts of the measurement operator relate are found below in Section 2.4.

*Remark 2.1.* As we have alluded to, one of the important advantages of our method over a vectorization-based approach is that the maps  $\mathcal{A}$  and  $\mathcal{A}_{2\text{nd}}$  require less memory to store than those considered in [56]. In particular,  $\mathcal{A}$  requires  $d'mn^\kappa$  entries to be stored in order to sketch a dimension of  $m^{d'}$  and  $\mathcal{A}_{2\text{nd}}$  requires  $d'mn^\kappa + m^{d'}m_{2\text{nd}}$  in order to sketch a dimension of  $m_{2\text{nd}}$ . By contrast, the map considered in [56] requires  $n^d m$  entries to reach a dimension of  $m$ . To make this more concrete, we consider the case where  $d = 4$ ,  $n = 40$ ,  $\kappa = 2$ , the final target dimension is 10,000, and our maps are dense matrices with Gaussian entries. In this case,  $\mathcal{A}_{2\text{nd}}$ , with intermediate dimensions of  $m_1 = m_2 = 250$  would require  $40^2 \times 250 \times 2 + 250^2 \times 10,000 = 625,800,000$  random entries, with a storage cost of about 2.5 gigabytes assuming 32-bit floating point arithmetic and the SI meaning of the prefix giga as  $10^9$  bytes. The vectorization based approach requires  $40^4 \times 10,000 = 25,600,000,000$  random entries at a storage cost of about 102.4 gigabytes. In Section 2.2, we will show that under these settings tensor recovery experiments using  $\mathcal{A}_{2\text{nd}}$  have an identical recovery rate reliability in both the Gaussian case and SORS case when compared to those that use vectorization-based compression, despite the 40 times smaller memory requirement.

## 2.2 Experiments

In this section, we show empirically that TIHT can be used with modewise measurement maps as defined in (2.2) to successfully reconstruct low-rank tensors. In our experiments we will consider randomly generated four-mode tensors in  $\mathbb{R}^{n \times n \times n \times n}$  for  $n = 40$  and  $n = 96$ . In the case where  $n = 40$ , we will utilize both modewise Gaussian and *RFD* measurements, see Definition 1.24. In the case where  $n = 96$  we will only consider *RFD* measurements, since generating and storing the dense Gaussian matrices takes up too much memory.

We compare our two-step modewise approach to a vectorization based method like was described in [56]. We consider the percentage of successfully recovered tensors from batches of 100 randomly generated low-rank tensors, as well as the average number of iterations used for recovery on the successful runs. In the case where  $n = 40$ , we consider the algorithm to have successfully

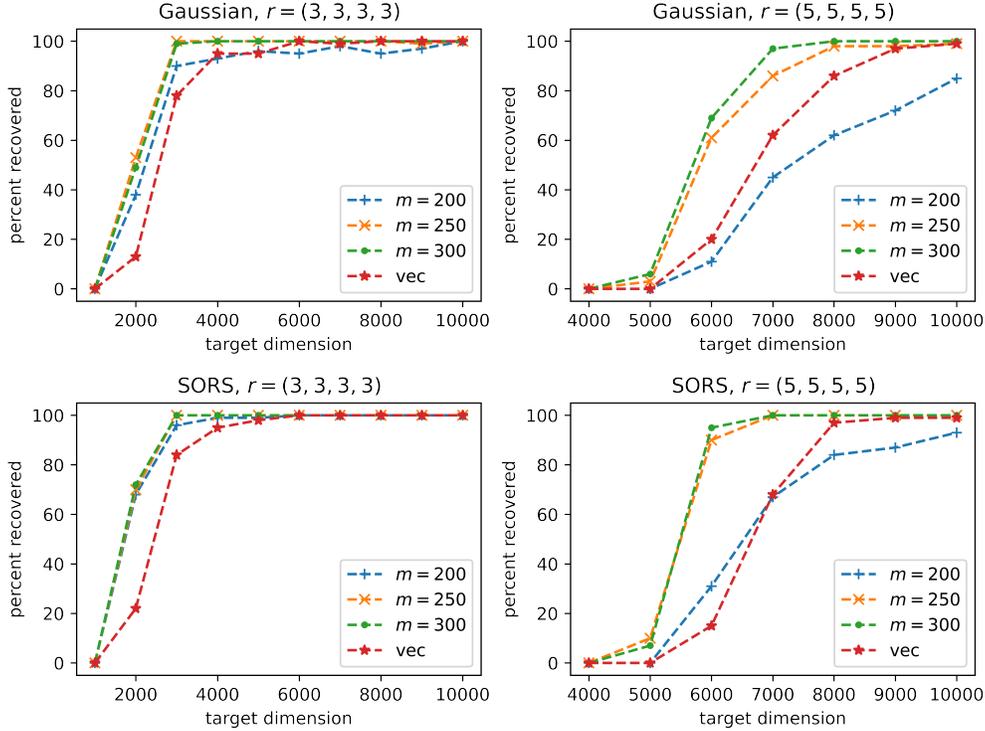


Figure 2.1 Fraction of successfully recovered random tensors out of a random sample of 100 tensors in  $\mathbb{R}^{40 \times 40 \times 40 \times 40}$  with various intermediate dimensions. A run is considered successful if relative error is below 1% in at most 1000 iterations.

recovered the tensor if the relative error falls below 1% in at most 1000 iterations. Similarly, in the case where  $n = 96$ , we consider the algorithm to have successfully recovered the tensor if the relative error falls below 5% in at most 1000 iterations. Compression, defined here as the ratio of the total number of entries in the measurements over total number of entries in the true tensor, ranges from about 0.04% to 0.4% depending on the choice for final sketching dimension. In all instances, we initialize the algorithm using a randomly generated low-rank tensor.

In our experiments, we apply the map from (2.9) with  $\kappa = 2$ . That is, we reshape a four-mode tensor whose modes are all of length  $n$  into a  $n^2 \times n^2$  matrix, perform modewise measurements reducing each of the two reshaped modes to  $m$ , the choice for intermediate dimension, and then in the second stage, vectorize that result and compress it further to  $m_{2\text{nd}}$ , the final target dimension. In our experiments, we consider a variety of intermediate dimensions to demonstrate the stability of advantage of the modewise measurements over the vectorized ones. In our experiments, for fair comparison, we will set the final embedding of our two-step process  $m_{2\text{nd}}$  equal to the final

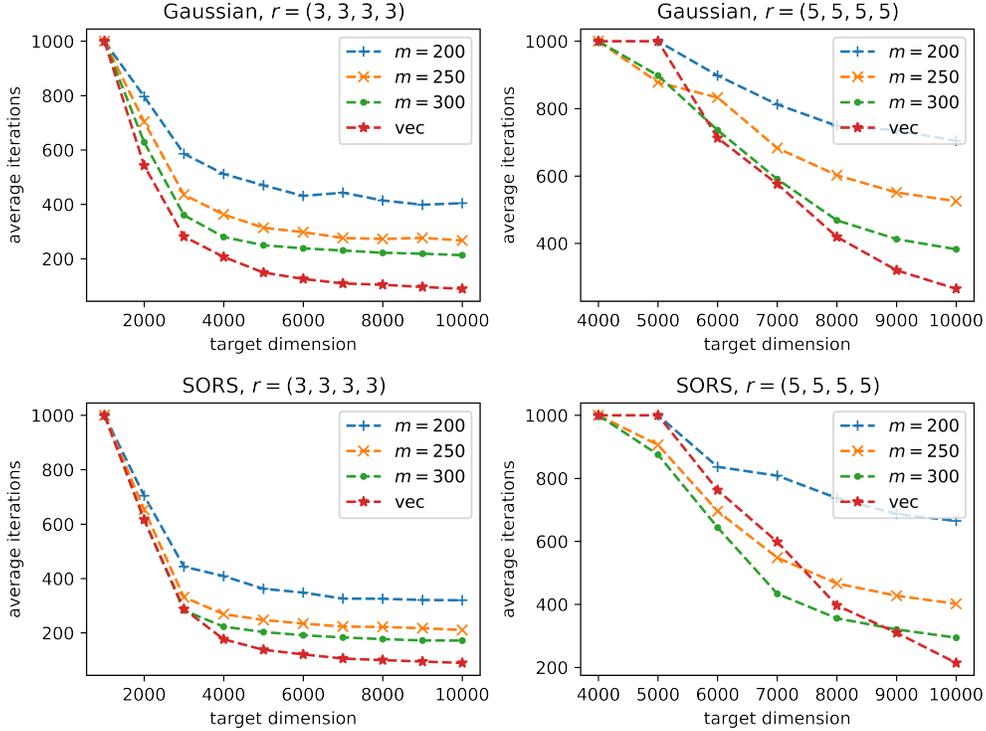


Figure 2.2 Average number of iterations until convergence among the successful runs out of a random sample of 100 tensors in  $\mathbb{R}^{40 \times 40 \times 40 \times 40}$  with various intermediate dimensions. A run is considered successful if relative error is below 1% in at most 1000 iterations.

embedding dimension of the vectorized method  $m_{\text{final}}$ . For a unified presentation, we will refer to the final dimension as  $m_0$  in either case - modewise or vectorized.

As noted in Remark 2.1, our proposed two-step method offers significant storage savings as compared to the vectorized based approach when looking at the size of the measurement maps involved. For example, when we use Gaussian measurements with  $m_0 = 10,000$  and  $n = 40$ , the vectorized computations had to be carried out using four NVIDIA v100 GPUs in parallel, whereas the two-step method easily fits on memory of one GPU. In the case where  $n = 96$ , generating, storing and applying Gaussian random matrices is impractical. For instance in the scenario we consider in Figure 2.3, where  $n = 96$  and  $m_0 = 65,536$ , we would need more than 22.25 terabytes to store the Gaussian map required for the vectorized approach. A two-step approach would require about 4.4 terabytes to store the measurement matrices.

We also note that we may apply *RFD* measurements more quickly to larger tensors than Gaussian measurements since the FFT enables *RFD* measurement matrices to be applied efficiently to the

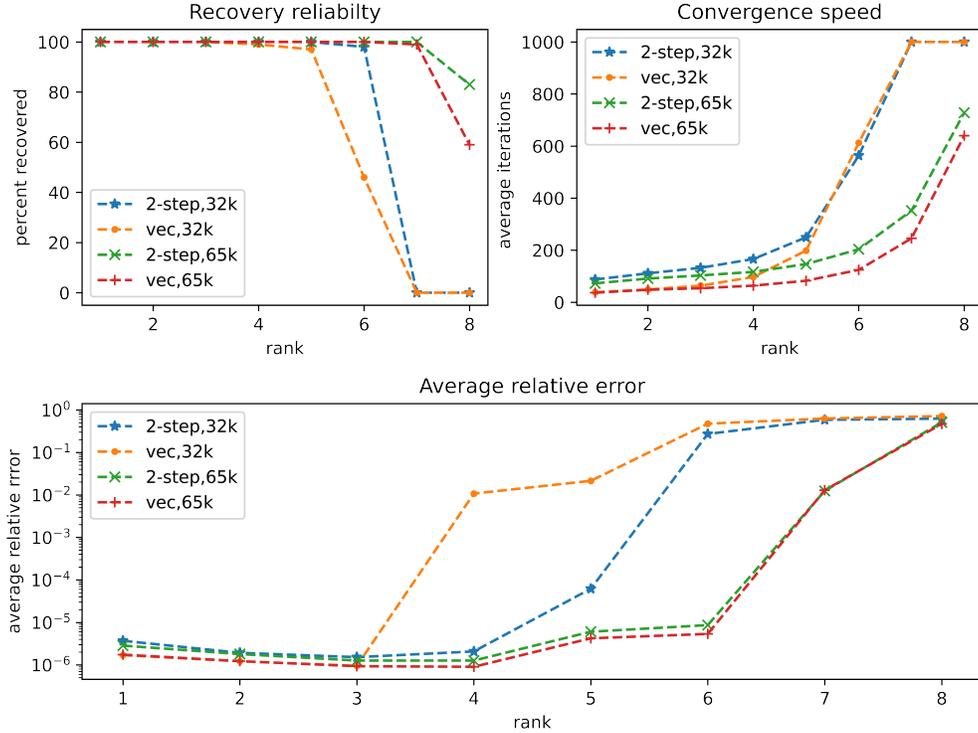


Figure 2.3 Top row shows recovery reliability and convergence speed for trials consisting of 100 samples of random tensors in  $\mathbb{R}^{96 \times 96 \times 96 \times 96}$  at various ranks. A run is successful if relative error reaches below 5% within 1000 iterations. Final target dimensions are  $m = 32,768$  and  $65,636$ , for two-step method intermediate dimensions are 2048 and 4096 respectively. Bottom row is average relative error over 100 samples after 500 iterations.

tensor and without the need to explicitly form all the measurement maps. In particular, we need only store the sign changes which form the diagonal of matrix  $D$ , and store the choice of which rows are sampled from  $F$  to form matrix  $RF$  (see Definition 1.24). For this reason, the same size of problem requires about 67 megabytes storage in two-step method and 340 megabytes for the vectorized approach when using  $RFD$  and an appropriately implemented FFT. It is for these reasons we restricted ourselves to  $RFD$  measurements for the  $n = 96$  case for both the vectorized and two-step approach.

As shown in Figures 2.1 and 2.2 the two-step approach, when compared to the vectorized approach with the same choice of final sketching dimension, shows reliable recovery, and comparable convergence rate with both Gaussian and  $RFD$  measurements. Indeed, for some choices of intermediate and final sketching dimensions, modewise measurements empirically recover low-rank tensors more reliably than vectorized measurements (see Figure 2.1, bottom row). We show that

these advantages do not result in the need for a substantially increased number of iterations in order to achieve our convergence criteria. Across all considered scenarios, the average number of iterations to meet the convergence criteria can be bounded by two to eight times the number needed in the corresponding vectorized approach, depending on the choice  $m$  for intermediate sketching dimension. Interestingly, for some ranges of the intermediate and final target dimension  $m_0$ , and in the case of *RFD* measurements in the rank  $(5, 5, 5, 5)$  instance, fewer iterations are needed (see Figure 2.2, bottom right). Thus, modewise measurements are an effective, memory-efficient method of dimension reduction, and the choice of intermediate sketching dimension allows us to further balance trade-offs in terms of convergence speed and overall memory requirements, given a particular size and rank.

In Figure 2.3, we investigate the performance of the algorithms as rank of the tensor increases. A larger tensor, with  $n = 96$  enables us to consider a wider range of  $\mathbf{r}$ 's that are reasonably considered to be low-rank. We maintain the vectorized to two-step comparison, and also consider two different final sketching dimensions,  $m_0 = 32, 768$  and  $65, 536$ . Due to the larger problem size and ranks, we scaled the convergence criteria to be 5% relative error in at most 1000 iterations for the comparison of recovery reliability and convergence speed. We observed in terms of performance of the algorithms a phase change as rank of the tensor increases for a fixed final sketching dimension. In particular, for  $m_0 = 32, 768$  the performance of vectorized and two-step approach empirically degrade significantly at  $r = 6$  and  $7$  respectively for this convergence criteria. When doubling the sketching dimension to  $m_0 = 65, 536$  we see empirically that phase change that the drop in performance occurs at  $r = 8$ . The bottom row of Figure 2.3 shows the shift in terms of average relative error for a fixed number of iterations. For the larger ranks and smaller sketching dimensions we observe that stagnation at non-global optima appears more likely and the runtimes required for acceptable recovery can become impractical. Both vectorized and two-step approaches have this feature, however for a fixed final sketching dimension, for some ranks near the threshold the two-step method performs incrementally better.

### 2.3 Discussion

As we have outlined in the previous section both theoretically and empirically, in terms of performance and economizing storage, using a two-stage measurement operator versus a one-stage vectorized only operator has significant advantages. There are a few shortcomings we wish to note however that came about as we developed the method. To understand the first one, consider the Corollary 2.2, which states that if the modewise transforms applied are assumed to be random matrices with i.i.d. sub-Gaussian entries with at least as many as (2.11) and (2.12) rows for the first and second stage respectively, then  $\mathcal{A}_{2nd}$  will have desired TRIP( $\delta, \mathbf{r}$ ) with high probability. That is, this is a result which shows that a certain class of random matrix will likely satisfy the TRIP, and thus will be useful in yielding measurements, should its sketching dimension size satisfy the following bounds.

**Corollary 2.2.** *Let  $r \geq 2$  and let  $\mathbf{r} = (r, r, \dots, r) \in \mathbb{R}^d$ . Suppose that  $\mathcal{A}$  and  $\mathcal{A}_{2nd}$  are defined as in (2.6) and (2.9), and that all of the  $A_i \in \mathbb{R}^{m \times n^\kappa}$  have i.i.d.  $\frac{1}{m}$ -subgaussian entries for all  $i = 1, \dots, d'$ , where  $d' = d/\kappa$ , and suppose that  $0 < \eta, \delta < 1$ . Let*

$$m \geq C\delta^{-2}r^{2d} \max \left\{ \frac{nd^2 \ln(\kappa)}{\kappa}, \frac{d^2}{\kappa^2} \ln \left( \frac{2d}{\kappa\eta} \right) \right\}, \quad (2.11)$$

and let  $A_{2nd} \in \mathbb{R}^{m_{2nd} \times m}$  be a  $\frac{1}{m_{2nd}}$ -sub-gaussian matrix with i.i.d. entries with

$$m_{2nd} \geq C\delta^{-2} \max \left\{ \left( \frac{r^d \kappa + dmr^\kappa}{\kappa} \right) \ln \left( \frac{d}{\kappa} + 1 \right) + \frac{dmr^\kappa}{\kappa} \ln \left( 1 + \delta r^d \right) + \frac{d^2 mr^\kappa \delta}{\kappa^2}, \ln \left( \frac{2}{\eta} \right) \right\}. \quad (2.12)$$

Then,  $\mathcal{A}_{2nd}$  satisfies the TRIP( $\delta, \mathbf{r}$ ) property, i.e.,

$$(1 - \delta)\|\mathcal{X}\|_2^2 \leq \|\mathcal{A}_{2nd}(\mathcal{X})\|_2^2 \leq (1 + \delta)\|\mathcal{X}\|_2^2,$$

for all  $\mathcal{X}$  with HOSVD rank at most  $\mathbf{r}$  with probability at least  $1 - \eta$ .

Observe that if  $\kappa = 1$ , then (2.11) implies that  $m > n$ , so the sketching dimension needs to be bigger than the original side length to ensure the error guarantee, and thus not accomplishing any compression. In other words, the theory we have stated here is silent about the case where we compress using modewise products without first combining at least some of the modes. Furthermore, numerical investigations lent empirical evidence that this may not merely have been an artifact

of the analysis, or a lack of tightness is the row bounds for this particular class of measurement operator. This was a disappointing state of affairs because it is quite natural to wish to apply no reshaping initially, and instead compress along all modes using modewise products before employing subsequent stages of reshaping and modewise linear transforms. Moreover, the necessity of having  $\kappa$  be at least two means that there may not be any natural or intuitive way to first reshape a tensor with say a prime number of modes; in that case some modes would have to be grouped and treated differently than others; which makes the choice of the operator data dependent in a way we should like to avoid.

The next two observations stem from the use of TIHT as a recovery procedure in the overall method. In the first sub-step of TIHT (2.2), we must calculate and form the intermediary tensor  $\mathcal{Y}^j$  which has the same dimensions as our original signal  $\mathcal{X}$ . Furthermore,  $\mathcal{Y}^j$  is emphatically not in a factorized form, nor would we expect it to be exactly low rank. This iterate is a full, un-factored, tensor that we have to store and operate on in practical terms that is the same size as the *uncompressed* data  $\mathcal{X}$ . In turn, in the next sub-step we then must find a factorization which approximates  $\mathcal{Y}^j$ . Naturally that factorization will be a significantly smaller object to store in memory in the low rank case.

Secondly, As was alluded too early in the Section 2.1, in Theorem 2.1, there is a reliance on an assumption about the quality of fit for the thresholding operator that exceeds the guarantees that we are aware of for the usual procedure to find the HOSVD. Furthermore calculating an HOSVD itself can qualify as computationally intensive; and refining a factorization could itself involve many iterations of say Algorithm 1.2 within an already iterative step of TIHT.

So the use of TIHT as the recovery procedure, and its reliance on the TRIP property introduces several significant downsides. It means we need to have working memory on the order of the size of the uncompressed, unfactorized data to perform the sub-step which is contraindicating the expressed goal of a low-memory (at all stages) method for tensor recovery. Ideally we would have some means to go from the compressed measurements  $\mathcal{L}(\mathcal{X})$  directly to core  $\mathcal{G}$  and factors  $U_i$  of the estimate without needing at any point in the procedure to form the entire, uncompressed tensor;

and thus remain low-memory throughout the entire process. Next, TIHT relies on an assumption about the thresholding operator  $\mathcal{H}_r$ ; and practically any effort taken to shore up that assumption may involve significantly more computational effort at each TIHT update, such as HOOI iterations. Lastly, all known methods to prove TRIP for two-stage measurement operators that use the often studied types of random modewise maps requires an initial reshaping; which may not be desirable or intuitive.

The desire to address these problems is what led us to investigate the Leave-One-Out alternative and its accompanying simple recovery procedure and is the subject of our next chapter.

## 2.4 Proof of Theorem 2.2

The goal of this section is to outline the proof of Theorem 2.2, which crucially builds on a similar type guarantee for the first stage compression operator and a lemma which shows how the output of the first stage of compression will likely belong in the set of nearly orthonogonal tensors as in Definition 2.4, and provides a covering number bound for this set. Once we have these, it is easy to fashion two-stage operators from standard types of random matrices, such as sub-Gaussian random matrices or *RFD* matrices, as was done e.g. Lemma 1.6. For a detailed proof of these results, see Section 5 and Appendices A and B of our work [23].

**Theorem 2.3.** *Suppose that  $\mathcal{A}$  is defined as per (2.6) and that each of the  $A_i$  have  $\text{RIP}(\epsilon, S_{1,2})$  property. Let  $r \geq 2$ , let  $\delta = 4d'r^d\epsilon$  and assume that  $\delta < 1$ . Then  $\mathcal{A} \circ \mathcal{R}$  satisfies the  $\text{TRIP}(\delta, \mathbf{r})$  property, i.e.,*

$$(1 - \delta)\|X\|_2^2 \leq \|\mathcal{A}(\mathcal{R}_1(X))\|_2^2 \leq (1 + \delta)\|X\|_2^2 \quad (2.13)$$

for all  $X$  with HOSVD rank less than  $\mathbf{r} = (r, r, \dots, r)$ .

So the goal is now to first prove this result about the first stage compression operator. To do so, it will be convenient to write  $\mathcal{A}$  as a composition of maps

$$\mathcal{A}(\mathcal{Y}) = \mathcal{A}_{d'}(\dots(\mathcal{A}_1(\mathcal{Y}))), \text{ where } \mathcal{A}_i(\mathcal{Y}) = \mathcal{Y} \times_i A_i \text{ for } 1 \leq i \leq d'. \quad (2.14)$$

We note that by (2.5), we may still write the reshaped  $d'$ -mode tensor  $\overset{\circ}{X}$  as a sum of  $r^d$  orthogonal tensors; that is by Lemma 1.4 we can write  $X$  as a sum of  $r^d$  rank one orthogonal tensors, and

the reshaping operator  $\mathcal{R}_1$  is linear. This action of a linear operator in a vector space working on orthogonal sums plays nicely with an approximation property; as we now state in Lemma 2.1

**Lemma 2.1.** *Let  $\mathcal{V}$  be an inner product space and let  $\mathcal{L}$  be a linear operator on  $\mathcal{V}$ . Let  $\mathcal{U} \subset \mathcal{V}$  be a subspace of  $\mathcal{V}$  spanned by an orthonormal system  $\{\mathbf{v}_1, \dots, \mathbf{v}_K\} \in \mathcal{V}$ . Suppose that*

$$(1 - \epsilon)\|\mathbf{v}_i\|_2^2 \leq \|\mathcal{L}\mathbf{v}_i\|_2^2 \leq (1 + \epsilon)\|\mathbf{v}_i\|_2^2 \quad \text{for all } 1 \leq i \leq K. \quad (2.15)$$

and also that

$$(1 - \epsilon)\|\mathbf{v}_i \pm \mathbf{v}_j\|_2^2 \leq \|\mathcal{L}(\mathbf{v}_i \pm \mathbf{v}_j)\|_2^2 \leq (1 + \epsilon)\|\mathbf{v}_i \pm \mathbf{v}_j\|_2^2 \quad \text{for all } 1 \leq i, j \leq K. \quad (2.16)$$

Then we have

$$(1 - K\epsilon)\|\mathbf{w}\|_2^2 \leq \|\mathcal{L}\mathbf{w}\|_2^2 \leq (1 + K\epsilon)\|\mathbf{w}\|_2^2 \quad \text{for all } \mathbf{w} \in \mathcal{U}.$$

*Proof.* Claim:

$$|\langle \mathcal{L}\mathbf{v}_i, \mathcal{L}\mathbf{v}_j \rangle| \leq \epsilon \quad \text{for all } i \neq j.$$

To prove the claim, let  $i \neq j$ . Then,

$$\begin{aligned} 4\langle \mathcal{L}\mathbf{v}_i, \mathcal{L}\mathbf{v}_j \rangle &= \|\mathcal{L}(\mathbf{v}_i + \mathbf{v}_j)\|_2^2 - \|\mathcal{L}(\mathbf{v}_i - \mathbf{v}_j)\|_2^2 \\ &\leq (1 + \epsilon)\|\mathbf{v}_i + \mathbf{v}_j\|_2^2 - (1 - \epsilon)\|\mathbf{v}_i - \mathbf{v}_j\|_2^2 \\ &= (1 + \epsilon)(\|\mathbf{v}_i\|_2^2 + \|\mathbf{v}_j\|_2^2 + 2\langle \mathbf{v}_i, \mathbf{v}_j \rangle) - (1 - \epsilon)(\|\mathbf{v}_i\|_2^2 + \|\mathbf{v}_j\|_2^2 - 2\langle \mathbf{v}_i, \mathbf{v}_j \rangle) \\ &= 4\langle \mathbf{v}_i, \mathbf{v}_j \rangle + 2\epsilon(\|\mathbf{v}_i\|_2^2 + \|\mathbf{v}_j\|_2^2) \\ &= 4\epsilon, \end{aligned}$$

where the last inequality follows from the fact that  $\|\mathbf{v}_i\|_2^2 = \|\mathbf{v}_j\|_2^2 = 1$  and  $\langle \mathbf{v}_i, \mathbf{v}_j \rangle = 0$ . Thus,  $\langle \mathcal{L}\mathbf{v}_i, \mathcal{L}\mathbf{v}_j \rangle \leq \epsilon$ . The reverse inequality is similar.

We now proceed to the Lemma itself, and argue by induction on  $K$ . When  $K = 1$ , the result is immediate from (2.15) and the fact that  $\mathcal{L}$  is linear. Now assume the result is true for  $K - 1$ . An arbitrary element of  $\mathcal{U}$  may be written as

$$\mathbf{w} = \sum_{i=1}^K c_i \mathbf{v}_i$$

where  $c_1, \dots, c_K$  are scalars. We will write  $\mathbf{w} = \mathbf{w}_{K-1} + c_K \mathbf{v}_K$ , where

$$\mathbf{w}_{K-1} := \sum_{i=1}^{K-1} c_i \mathbf{v}_i.$$

By construction, we have

$$\|\mathcal{L}\mathbf{w}\|_2^2 = \|\mathcal{L}\mathbf{w}_{K-1}\|_2^2 + \|c_K \mathcal{L}\mathbf{v}_K\|_2^2 + 2c_K \langle \mathcal{L}\mathbf{w}_{K-1}, \mathcal{L}\mathbf{v}_K \rangle.$$

We may use the inequality  $2ab \leq a^2 + b^2$  along with the claim earlier to see

$$\begin{aligned} 2c_K \langle \mathcal{L}\mathbf{w}_{K-1}, \mathcal{L}\mathbf{v}_K \rangle &= \sum_{i=1}^{K-1} 2c_i c_K \langle \mathcal{L}\mathbf{v}_i, \mathcal{L}\mathbf{v}_K \rangle \\ &\leq \epsilon \sum_{i=1}^{K-1} 2c_K c_i \\ &\leq \epsilon \sum_{i=1}^{K-1} (c_K^2 + c_i^2) \\ &\leq (K-1)\epsilon c_K^2 + \epsilon \sum_{i=1}^{K-1} c_i^2. \end{aligned} \tag{2.17}$$

By the inductive assumption,

$$\|\mathcal{L}\mathbf{w}_{K-1}\|_2^2 \leq (1 + (K-1)\epsilon) \|\mathbf{w}_{K-1}\|_2^2 = (1 + (K-1)\epsilon) \sum_{i=1}^{K-1} c_i^2.$$

Thus,

$$\begin{aligned} \|\mathcal{L}\mathbf{w}\|_2^2 &\leq (1 + (K-1)\epsilon) \sum_{i=1}^{K-1} c_i^2 + (1 + \epsilon)c_K^2 + (K-1)\epsilon c_K^2 + \epsilon \sum_{i=1}^{K-1} c_i^2 \\ &= (1 + K\epsilon) \sum_{i=1}^K c_i^2 = (1 + K\epsilon) \|\mathbf{w}\|_2^2. \end{aligned} \tag{2.18}$$

The reverse inequality is similar. □

The next lemma checks that, if  $A_i$  the matrix satisfies  $\text{RIP}(\epsilon, S_{1,2})$  property for some  $1 \leq d'$  then the operator  $\mathcal{A}_i$  as in (2.14) will satisfy the conditions of Lemma 2.1 for the system of rank one component tensors that are produced by our reshaping procedure.

**Lemma 2.2.** *Let  $\{\mathcal{V}_1, \dots, \mathcal{V}_K\} \in \mathbb{R}^{n \times \dots \times n}$  be an orthonormal system of rank one tensors of the form  $\mathcal{V}_k = \bigcirc_{i=1}^{d'} \mathbf{v}_k^i$  where  $\|\mathbf{v}_k^i\| = 1$  for all  $1 \leq i \leq d'$ . Let  $1 \leq i \leq d'$ , suppose  $A_i$  has the*

RIP( $\epsilon/2, S_{1,2}$ ) property and assume that each  $\mathbf{v}_k^i$  is an element of the set  $S_1$  defined in Definition 2.3. Then the conditions (2.15) and (2.16) are satisfied for (the vectorizations of) these  $\{\mathcal{V}_i\}_{i=1}^K$  and  $\mathcal{L} = \mathcal{A}_i$  defined via  $\mathcal{A}_i(\mathcal{X}) = \mathcal{X} \times_i A_i$ .

The next lemma gives a formula for the tensor  $\mathcal{Y}_t$  obtained by applying sequentially the first  $t$  of the maps  $\mathcal{A}_i$ . It shows that all intermediary tensors  $\mathcal{Y}_t$  can be written as an orthogonal linear combination of  $r^{\kappa(d'-t)}$  rank-one tensors of unit norm. Moreover, for each of the terms in this sum, the  $(t+1)$ -st component vector is  $\mathring{\mathbf{u}}_{j_{t+1}}^{(t+1)}$  as defined in (2.5) and therefore is an element of the set  $S_1$ .

**Lemma 2.3.** *Let  $\mathcal{Y}_0 = \mathring{\mathcal{X}}$  and  $\mathcal{Y}_t := \mathcal{A}_t(\mathcal{Y}_{t-1}) = \mathcal{Y}_{t-1} \times_t A_t$  for all  $t = 1, \dots, d'$ . Then, for each  $1 \leq t \leq d' - 1$ , we may write*

$$\mathcal{Y}_t = \sum_{j_{d'}=1}^{r^\kappa} \cdots \sum_{j_{t+1}=1}^{r^\kappa} \mathcal{G}_{t(j_{t+1}, \dots, j_{d'})} \left[ \left( \bigcirc_{i=1}^t \mathbf{v}_{j_{t+1}, \dots, j_{d'}}^{(i)} \right) \circ \left( \bigcirc_{i=t+1}^{d'} \mathring{\mathbf{u}}_{j_i}^{(i)} \right) \right], \quad (2.19)$$

where  $\|\mathbf{v}_{j_{t+1}, \dots, j_{d'}}^{(i)}\| = 1$  for all valid index subsets.

We are now ready to prove Theorem 2.3.

*Proof of Theorem 2.3.* By Lemma 1.4, we can write  $\mathcal{Y}_0 = \mathring{\mathcal{X}}$  as an orthogonal linear combination of  $r^d$  norm one terms of the form

$$\bigcirc_{i=1}^{d'} \mathring{\mathbf{u}}_{j_i}^{(i)}, \quad 1 \leq j_i \leq r^\kappa,$$

where each of the vectors  $\mathring{\mathbf{u}}_{j_i}^{(i)}$ ,  $1 \leq j_i \leq r^\kappa$ , are obtained as the vectorization of a rank-one  $\kappa$ -mode tensor. Therefore, since  $A_1$  satisfies RIP( $\epsilon, S_{1,2}$ ), Lemma 2.2 allows us to apply Lemma 2.1 to see

$$\|\mathcal{A}_1(\mathring{\mathcal{X}})\|_2 \leq (1 + 2r^d \epsilon) \|\mathring{\mathcal{X}}\|_2. \quad (2.20)$$

Next observe that by Lemma 2.3

$$\mathcal{Y}_t = \overbrace{\sum_{j_{d'}=1}^{r^\kappa} \cdots \sum_{j_{t+1}=1}^{r^\kappa}}^{(d'-t) \text{ sums of } r^\kappa \text{ terms each}} \mathcal{G}_{t(j_{t+1}, \dots, j_{d'})} \overbrace{\left[ \left( \bigcirc_{i=1}^t \mathbf{v}_{j_{t+1}, \dots, j_{d'}}^{(i)} \right) \circ \left( \bigcirc_{i=t+1}^{d'} \mathring{\mathbf{u}}_{j_i}^{(i)} \right) \right]}^{\text{remains a set of orthonormal vectors}}$$

and note that there are  $r^{\kappa(d'-t)}$  terms appearing in the sum in (2.19). Therefore, Lemmas 2.1 and 2.2 allow us to see that

$$\|\mathcal{Y}_{t+1}\|_2 \leq \left(1 + 2r^{\kappa(d'-t)} \epsilon\right) \|\mathcal{Y}_t\|_2 \quad (2.21)$$

for  $1 \leq t \leq d' - 1$ . Since  $\mathcal{Y}_{d'} = \mathcal{A}(\overset{\circ}{\mathcal{X}})$ , combining (2.20) and (2.21) for each of the compositions implies that the overall first stage compression operator  $\mathcal{A}$  defined in (2.6) satisfies

$$\|\mathcal{A}(\overset{\circ}{\mathcal{X}})\|_2 \leq \prod_{t=0}^{d'-1} \left(1 + 2r^{\kappa(d'-t)}\epsilon\right) \|\overset{\circ}{\mathcal{X}}\|_2.$$

To complete the upper bound set  $\alpha := 2r^d\epsilon$  and note that  $2\alpha < 1$ . Recall  $d' = \frac{d}{\kappa}$ . Then, since  $r \geq 2$

$$\begin{aligned} \prod_{t=0}^{d'-1} \left(1 + 2r^{\kappa(d'-t)}\epsilon\right) &= \prod_{t=0}^{d'-1} (1 + \alpha r^{-t\kappa}) \\ &= 1 + \alpha \sum_{t=0}^{d'-1} r^{-t\kappa} + \alpha^2 \sum_{\substack{t_1, t_2=0: \\ t_1 < t_2}}^{d'-1} r^{-(t_1+t_2)\kappa} + \dots + \alpha^{d'} r^{-(1+\dots+(d'-1))\kappa} \\ &\leq 1 + \alpha \sum_{t=0}^{d'-1} r^{-t\kappa} + \left(\alpha \sum_{t=0}^{d'-1} r^{-t\kappa}\right)^2 + \dots + \left(\alpha \sum_{t=0}^{d'-1} r^{-t\kappa}\right)^{d'} \\ &\leq 1 + \alpha \sum_{t=0}^{\infty} 2^{-t} + \left(\alpha \sum_{t=0}^{\infty} 2^{-t}\right)^2 + \dots + \left(\alpha \sum_{t=0}^{\infty} 2^{-t}\right)^{d'} \\ &\leq 1 + 2\alpha + (2\alpha)^2 + \dots + (2\alpha)^{d'} \\ &\leq 1 + 2d'\alpha \\ &= 1 + 4d'r^d\epsilon \end{aligned}$$

which completes the proof of the upper bound. The proof of the lower bound is nearly identical.  $\square$

Now that we have proved Theorem 2.3, and have now the first stage compression operator,  $\mathcal{A}(\mathcal{R}_1(\cdot))$  has a TRIP so long as the constituent linear maps themselves have a RIP, we can use this show that the second stage compression operator also as a TRIP. Unfortunately, we cannot simply inductively apply the same reasoning, since crucially the first stage depended on using that the sum of orthogonal (rank-1) components and a linear operator were compatible with the embedding properties. After the first stage, these sums of rank-1 tensors will almost certainly no longer be exactly orthogonal. Nevertheless, we see that they will be *nearly* orthogonal, and should we have a covering number for this set, then it will be possible to find a second stage compression operator which is able to satisfy the RIP for this set. This is what motivates the use of Definition 2.4, and, along with a covering number argument for this set of nearly orthogonal tensors, is the main content of the following Lemma.

**Lemma 2.4.** *Let  $\mathcal{X}$  be a unit norm  $d$ -mode tensor with HOSVD rank at most  $\mathbf{r} = (r, \dots, r)$ . Let  $\mathcal{A}$  be defined as in (2.6), assume that the matrices  $A_i$  have the RIP( $\epsilon, S_{1,2}$ ) property for all  $1 \leq i \leq d'$ , and that  $\delta = 12d'r^d\epsilon < 1$ . Then, the  $d'$ -mode tensor  $\mathcal{A}(\overset{\circ}{\mathcal{X}}) \in \mathbb{R}^{m \times \dots \times m}$  is an element of the set  $\mathcal{B}_{1+\epsilon, \epsilon, 1-\delta/3, \mathbf{r}'}$  (as per Definition 2.4). Additionally, let  $\tilde{\mathcal{B}}_{\epsilon, \delta, r} := \left\{ \frac{\mathcal{X}}{\|\mathcal{X}\|_F} \mid \mathcal{X} \in \mathcal{B}_{1+\epsilon, \epsilon, 1-\delta/3, \mathbf{r}'} \right\}$  and suppose that  $m \geq r^{d-1}$ . Then,*

$$C_t^{\|\cdot\|_2}(\tilde{\mathcal{B}}_{\epsilon, \delta, r}) \leq \left( \frac{9(d/\kappa + 1)}{t} \right)^{r^d + r^\kappa m d / \kappa} \left( (1 + \epsilon)^2 + \epsilon r^d \right)^{d m r^\kappa / \kappa} (1 + \epsilon)^{d^2 m r^\kappa \kappa^{-2}}$$

holds for all  $t > 0$ , and  $1 > \delta > 0$ .

Where  $C_t^{\|\cdot\|_2}(\tilde{\mathcal{B}}_{\epsilon, \delta, r})$  refers to the covering number, see Definition 1.27. The proof then of Theorem 2.2 follows from applying the RIP assumption of  $A_{2nd}$  after using Theorem 2.3 for  $\mathcal{A}(R_1(\mathcal{X}))$ .

## CHAPTER 3

### LEAVE ONE OUT MEASUREMENTS FOR RECOVERY

Our investigation of two-stage measurement operators that satisfy the TRIP was motivated by our desire to employ a result regarding TIHT that could, with important qualifications, solve the tensor recovery problem by using a memory efficient  $\mathcal{L} : \mathbb{R}^{n_1 \times \dots \times n_d} \rightarrow \mathbb{R}^{m_{final}}$  measurement operator. In Section 2.3 we noted three significant drawbacks to this overall approach

1. the method was not low-memory at all stages since a sub-step of TIHT involves forming and operating on a tensor the size of the uncompressed original signal
2. the method relies on an effectively unverifiable assumption about the thresholding operator that is also computationally expensive to shore up
3. the apparent necessity of an initial reshaping in order to achieve useful compression

The desire to address these issues led us to a useful specialization to the modewise measurement framework, which we refer to as the leave-one-out alternative.

We are not the first to propose the idea. Most notably, in the work of Hendrikx and De Lathauwer [26], they describe an algorithm that recovers a tensor from linear combinations of its entries where the measurement operations themselves are constrained to be Kronecker-structured; this will serve as a particular example which fits very naturally into our framework once we properly situate it. In that work they give necessary and sufficient conditions for perfect recovery of exactly low-rank tensors. They describe some relevant heuristics and provide empirical results for the performance of the recovery when the tensor or its measurements are subject to white noise. Additionally they demonstrate how to adapt their method to two other tensor formats, namely CP and tensor train.

Furthermore, In the work of Tropp and Udell et al [60], a comparable recovery procedure as [26] was described (in that work, Algorithm 4.1, Tucker Sketch) however the measurement ensembles analyzed were in some sense, less structured. Measurements used to estimate factors of a tensor in this work were conceived as operating on the unfoldings of the tensor, not strictly modewise as is possible in the Kronecker-structured case of [26]. This too we can situate into our framework from Section 1.5. In the work [60] the authors detail approximation error expectations

of the recovered tensor in terms quasi-optimality, and notably do not require the signal to be exactly low-rank, both things we find desirable in a theory on recovery. On the other hand, the error guarantees in that work are in *expectation* and are stated using the assumption that all the entries of the measurement matrices are sampled from independent Gaussian random variables; which from a practical view point can be undesirable as these measurement matrices will take up as much storage in a way that scales data tensor itself. The authors do empirically investigate the performance of a more structured and constrained measurement map, specifically when the sketching matrices are constructed by using Khatri-Rao products of Gaussian matrices. In those experiments they show that these structured measurements deliver nearly as good approximations at the benefit of using potentially significantly less space, since it is possible in this scenario to store only the parts and construct the ensemble as it is needed during sketching.

Our contribution in this chapter is to unify and add to these works. Using the overall same strategy as [60], our error analysis is applied to the more structured Kronecker and Khatri-Rao measurement ensembles which are then used in a Canonical Recovery Algorithm 3.1 to recover the factorized form of a tensor. These types of measurement ensembles operate on the tensor in subtly different modewise manner that we will situate in terms of the framework introduced in Section 1.5. In a similar fashion as to the type of measurement operator that was the subject of Chapter 2, this has significant practical advantages since neither the entire signal tensor nor the entire measurement operator need to be kept in working memory in order to obtain measurements or importantly, to recover the factors of the original signal tensor. The improvement compared to [26] then is that the error analysis presented in this chapter applies in the non-exact low-rank case, and quantifies more specifically how the error depends on the relevant parameters of the problem, such as rank truncation. Additionally, besides applying the analysis to these more structured measurement ensembles that was not present in those early works, we remove from it the needed assumption that the entries of the sketching matrices are drawn from Gaussian distribution independently, and instead rely on a Johnson-Lindenstrauss property (see Definition 1.18) in order to state *probabilistic* bounds on errors of recovered tensors. The advantage to this is that there are several well known distributions

of random matrices which are known to satisfy this property, and it becomes a straightforward exercise then to reinterpret measurement requirements for different choices of sketching matrices. Having these options enables a user to select structured measurement schemes which can take advantage of fast matrix-vector multiplies for example, see Definition 1.24.

### 3.1 Kronecker Products, Khatri-Rao Products, and Leave-One-Out Modewise Measurement Notation and Examples

We now will describe and provide some notation and terminology for the type of measurement operators studied throughout this chapter. The general concept we are able to specialize to several cases is the following:

**Definition 3.1** (Leave-one-out Measurements). Given a  $d$ -mode tensor  $\mathcal{X}$  with side-lengths  $n$ , leave-one-out measurements are the result of reducing all but one mode using linear operations on the unfolding of the tensor. That is,

$$B_j = \Omega_{(j,j)} X_{[j]} \Omega_{-j}^T \quad (3.1)$$

are leave-one-out measurements whenever  $\Omega_{(j,j)} \in \mathbb{R}^{n \times n}$  and full-rank,  $\Omega_{-j} \in \mathbb{R}^{m' \times n^{d-1}}$  where  $m' \leq n^{d-1}$ .

Any measurement process that can be equivalently described as left multiplication of an unfolded tensor by a full-rank square matrix and right multiplied by some other linear operator that reduces all other modes is a leave-one measurement process. We will consider three specific cases for how to construct the overall measurement operator  $\Omega_{-j}$  in Definition 3.1: Kronecker structured, Khatri-Rao structured and unstructured measurement operators, and then situate this within the framework introduced in Section 1.5.

#### 3.1.1 Kronecker Structured Measurements

Kronecker structured leave-one-out measurements are constructed by taking the Kronecker product (see Definition 1.11) of  $d-1$  component matrices to fashion  $\Omega_{-j}$ , in addition to one square, full-rank measurement matrix  $\Omega_{(j,j)}$  to be applied to the mode- $j$  which is uncompressed. Note this matrix could simply be the identity matrix. We will require leave-one-out measurements of type in the Definition 3.1 for each mode (i.e.  $\forall j \in [d]$ ) in order to recover all factors. Collectively, the

matrices needed to define this ensemble will form a set  $\{\Omega_{(i,j)}\}_{i,j \in [d]}$  where  $\Omega_{(i,j)} \in \mathbb{R}^{m \times n}$  when  $j \neq i$ , and where  $\Omega_{(i,i)} \in \mathbb{R}^{n \times n}$  for all  $i \in [d]$ .

So far our measurements then are

$$\mathcal{B}_j := \mathcal{X} \underset{i=1}{\overset{d}{\times}} \Omega_{(j,i)} \quad (3.2)$$

for all  $j \in [d]$  using the matrices  $\{\Omega_{(j,i)}\}_{i=1}^d$ . Crucially, we can identify this tensor of measurements  $\mathcal{B}_j$  with a flattened version which makes it clear that Kronecker structured modewise measurements do define leave-one-out measurements conforming to Definition 3.1, see Lemma 1.2.

$$B_j = \Omega_{(j,j)} X_{[j]} \Omega_{-j}^T = \Omega_{(j,j)} X_{[j]} (\Omega_{(j,1)} \otimes \Omega_{(j,2)} \otimes \dots \otimes \Omega_{(j,j-1)} \otimes \Omega_{(j,j+1)} \otimes \dots \otimes \Omega_{(j,d)})^T \quad (3.3)$$

See Algorithm 3.2 for an outline of the sketching procedure. Note, (3.2) makes it clear we have here a one-stage modewise measurement process which has *no* initial reshaping; but instead leaves one mode uncompressed. As mentioned, we will require a collection of such measurements - one measurement tensor in one mode will be used only to recover a single factor matrix in the Tucker decomposition of our estimate. We refer to a collection of measurements from different measurement operators as a measurement ensemble.

Since it will be relevant in our analysis, we observe that conceptually, the Kronecker products of matrices can be rewritten in terms of matrix products of auxiliary matrices. This will be useful if we wish to examine via, e.g., (1.4), how several modewise products change the properties of the resulting factor matrices  $U_j$  in the Tucker decomposition of a given tensor. As an illustration, consider the case of a three mode tensor  $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ . Let  $\mathbf{x} \in \mathbb{R}^{n_1 n_2 n_3}$  denote the vectorization of  $\mathcal{X}$ , and further suppose that  $\Omega_j \in \mathbb{R}^{m_j \times n_j}$  for  $j = 1, 2, 3$  are three measurement matrices used to produce modewise measurements of  $\mathcal{X}$  given by  $\mathcal{X} \times_1 \Omega_1 \times_2 \Omega_2 \times_3 \Omega_3$ . Allowing for an additional reshaping, we can identify these three modewise operations equivalently with a single matrix-vector product using a variant of (1.4). That is, one can see that  $\text{vec}(\mathcal{X} \times_1 \Omega_1 \times_2 \Omega_2 \times_3 \Omega_3) = \Omega \mathbf{x}$  where  $\Omega = \Omega_3 \otimes \Omega_2 \otimes \Omega_1 \in \mathbb{R}^{m_1 m_2 m_3 \times n_1 n_2 n_3}$ . Let  $I_n$  denote the  $n \times n$  identity matrix. The mixed-product

property of the Kronecker product can now be used to further show that in fact  $\Omega = \tilde{\Omega}_3 \tilde{\Omega}_2 \tilde{\Omega}_1$ , where

$$\begin{aligned}\tilde{\Omega}_1 &= I_{n_3} \otimes I_{n_2} \otimes \Omega_1 \in \mathbb{R}^{m_1 n_2 n_3 \times n_1 n_2 n_3}, \\ \tilde{\Omega}_2 &= I_{n_3} \otimes \Omega_2 \otimes I_{m_1} \in \mathbb{R}^{m_1 m_2 n_3 \times m_1 n_2 n_3}, \\ \tilde{\Omega}_3 &= \Omega_3 \otimes I_{m_2} \otimes I_{m_1} \in \mathbb{R}^{m_1 m_2 m_3 \times m_1 m_2 n_3}.\end{aligned}$$

Hence,  $\text{vec}(\mathcal{X} \times_1 \Omega_1 \times \Omega_2 \times_3 \Omega_3) = \tilde{\Omega}_3 \tilde{\Omega}_2 \tilde{\Omega}_1 \text{vec}(\mathcal{X})$ . This example can be easily be extended to any number of modes.

### 3.1.2 Khatri-Rao Structured Measurements

Leave-one-out measurement ensembles which are Khatri-Rao products of smaller maps are also possible and have been considered before in works such [60]. That is,

$$\begin{aligned}B_j &= \Omega_{(j,j)} X_{[j]} \Omega_{-j}^T \\ &= \Omega_{(j,j)} X_{[j]} \left( \Omega_{(j,1)}^T \odot \Omega_{(j,2)}^T \odot \dots \odot \Omega_{(j,j-1)}^T \odot \Omega_{(j,j+1)}^T \dots \odot \Omega_{(j,d)}^T \right)^T \\ &= \Omega_{(j,j)} X_{[j]} \left( \Omega_{(j,1)} \cdot \Omega_{(j,2)} \cdot \dots \cdot \Omega_{(j,j-1)}^T \cdot \Omega_{(j,j+1)} \dots \cdot \Omega_{(j,d)} \right)^T\end{aligned}\quad (3.4)$$

where  $\Omega_{(j,j)}$  is a full-rank square matrix, and  $\Omega_{(j,i)} \in \mathbb{R}^{m \times n}$  for  $j \neq i$ . Note that in this case we can consider  $\Omega_{-j} \in \mathbb{R}^{m \times n^{d-1}}$  as sketching sub-tensors of size  $n^{d-1}$  to size  $m$  (as opposed to  $m^{d-1}$  as was done previously with Kronecker structured measurement maps). See Algorithm 3.3 for an outline of the sketching procedure. Note, to situate this measurement process within the framework, unlike the Kronecker structured measurements, there is no simple equivalence of each component map to a single modewise product along a mode - instead we conceive of the measurement matrix acting on sub-tensors of  $d - 1$  modes - i.e. the subtensors that occupy the rows of a mode- $j$  unfolding. That is, there is a reshaping into a mode- $j$  flattening initially, and a single ‘‘modewise’’ (matrix) product of that reshaped tensor with the measurement operator. Fundamentally, measurements structured this way combine  $d - 1$  modes initially with a reshaping; however the full measurement matrix itself need not be fully pre-computed because of its Khatri-Rao structure. One can, at the time of sketching, take the Kronecker product of the columns of the component matrices; trading compute for storage.

### 3.1.3 Unstructured Measurements

In [50], a type of leave-one-out measurements are described and analyzed, however in that work their measurement ensembles are not structured as was described in the previous two sections, but instead are dense matrices where  $\Omega_{-j} \in \mathbb{R}^{m \times n^{d-1}}$  has entries all drawn independently from a Gaussian distribution and this matrix is right multiplied by an unfolding of the tensor:

$$B_j = \Omega_{(j,j)} X_{[j]} \Omega_{-j}^T$$

where again  $\Omega_{(j,j)}$  is some square, full-rank matrix.

As an important observation, the storage requirements for the sketching matrix itself in this case is large, comparable in size to the data itself since the long dimension is  $n^{d-1}$ , which can be undesirable (though an improvement on the fully vectorized approach which must need have  $n^d$  as its long dimension). Furthermore, the error analysis in [50] conducted on this type of measurement when the unstructured matrix  $\Omega_{-j}$  has i.i.d. Gaussian entries relies on expectation bounds known for Gaussian matrices and their pseudo-inverses. Comparable bounds for other distributions, or for matrices constructed using Kronecker or Khatri-Rao products are not covered in the analysis. Note, Kronecker or Khatri-Rao structured measurements alleviate to a large degree the storage problem associated with  $\Omega_{-j} \in \mathbb{R}^{m \times n^{d-1}}$  when compared to in the unstructured case since the entire sketching matrix does not need to be maintained in memory, but rather just constituent components  $\Omega_{(j,i)} \in \mathbb{R}^{m \times n}$  can be stored, and the action of the Kronecker or Khatri-Rao product on the tensor can be computed as part of the sketching phase. This does incur additional operations in the sketching phase, and in our runtime analysis and numerical experiments we detail the trade-off between space and run-time for these various choices of measurement operators.

### 3.1.4 Core Measurements

Regardless of which type leave-one-measurements are used for each of the modes and recovering the factors, in order to recover the core factor using only a single access to the data  $\mathcal{X}$ , we will require an additional set of measurements. These are in all cases discussed in this chapter, modewise, and can compress *all* modes; i.e. the operator is a single stage, no initial reshaping

modewise measurement operator.

$$\mathcal{B}_c := \mathcal{X} \underset{j=1}{\overset{d}{\times}} \Phi_j \quad (3.5)$$

using the matrices  $\{\Phi_j\}_{j=1}^d$ . See Algorithm 3.4 for an outline of the sketching procedure for the core.

All together then, these  $d + 1$  measurement tensors,  $d$  of the leave-one-out type (Definition 3.1) and one of the type in (3.5) will be used to recover the parts of the original full tensor. Figure 3.1 is a schematic rendition of the overall measurement procedure in the three mode case ( $d = 3$ ).

Note that the  $n^d$  entries of the original tensor  $\mathcal{X}$  are compressed into  $dnm^{d-1} + m_c^d$  entries of the  $d + 1$  total different measurement tensors; one leave-one-out measurement for each of the  $d$  modes as well as the one measurement tensor for use in estimating the core. Recovery naturally will require the storage of the measurement operators themselves in some fashion along with the measurement tensors. As dense matrices, the measurement ensemble collectively uses  $d((d - 1)mn + n^2) + dm_c n$  total entries. However the number of random variables required to generate these matrices may be significantly fewer depending on the method employed. For example, when using *RFD* matrices and an FFT algorithm, the number of random bits required to construct an ensemble is linear in  $n$ . Additionally, some choices of measurement matrices allow for near-linear time matrix-vector multiplication.

As we detail later, applying the measurement operators, i.e. the sketching phase, is asymptotically the most computationally intensive part of the overall measure and recovery procedure, and thus in settings where it is useful to economize the computational effort to obtain measurements these matrices have significant advantages. Whichever choice for type of measurements are used, however, the efficiency in terms of the run-time of the recovery algorithm is largely dependent on the ratios  $m/n$  and  $m_c/n$ .

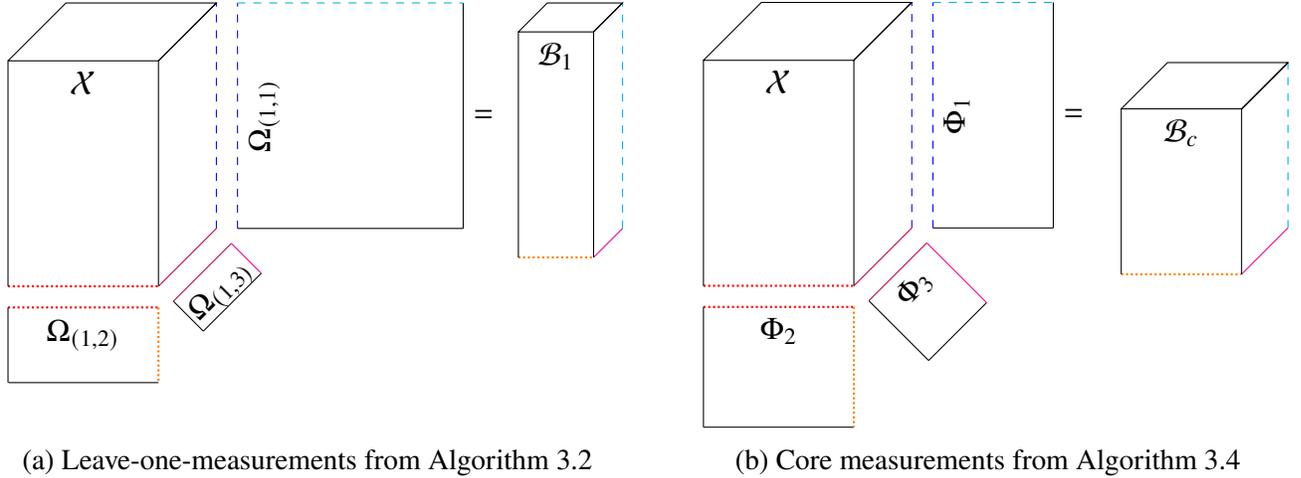


Figure 3.1 Schematic for the two types of measurement tensors used in recovery Algorithm 3.9 for a three mode tensor. Measurement tensors  $\mathcal{B}_i$  of the type shown in (a) for  $i = 1$  will be used to estimate the factors of the Tucker decomposition, and the measurement tensor  $\mathcal{B}_c$  in (b) will be used to estimate the core of the Tucker decomposition.

### 3.1.5 The Canonical One-Pass Recovery Algorithm

We can now state Algorithm 3.1 as the canonical algorithm for one-pass recovery the tensor using leave-one-out measurements. The algorithm outputs an estimate in factored form  $\mathcal{X}_1 = [[\mathcal{H}, Q_1, \dots, Q_d]]$ , and we say one-pass to emphasize that after measuring the tensor  $\mathcal{X}$ , no additional access to the data is required to obtain the estimate  $\mathcal{X}_1$ . The inputs to the algorithm are leave-one-out measurements  $\mathcal{B}_i$  for each of the modes of any type, as well as measurements  $\mathcal{B}_c$  for use in recovering the core, some of the related measurement operators  $\Omega_{(i,i)}$ ,  $\Phi_i$ , and a target rank  $\mathbf{r}$  parameter. Algorithm 3.1 consists of two loops. The first loop recovers the factor matrices  $Q_i$  in the Tucker factorization. The second loop depends on the output of the first loop and computes the core  $\mathcal{H}$  of the Tucker factorization. Importantly, at no point does a full, uncompressed tensor or size  $\mathcal{X}$  ever need to be formed. The procedure goes directly from measurements to factors.

Note that within the pseudo-code for Algorithm 3.1 the “unfold” function in the body of Algorithm 3.1 takes a tensor and flattens it into the given shape by arranging the specified mode’s fibers as the columns, e.g.  $H \leftarrow \text{unfold}(\mathcal{B}_c, m_c \times m_c^{d-1}, \text{mode} = 1)$  is equivalent to  $H \leftarrow (\mathcal{B}_c)_{[1]}$  using the notation of Section 1.4.1, the matrix that is obtained by flattening the  $d$ -mode tensor  $\mathcal{B}_c$  along mode-1. The “fold” function is the inverse of “unfold”, and takes a matrix that is an unfolding

along the specified mode of a tensor and reshapes it into a tensor with the specified and compatible dimensions, e.g.  $\mathcal{B} \leftarrow \text{fold}(H, r \times \underbrace{m_c \times \cdots \times m_c}_{d-1}, \text{mode} = 1)$  takes the matrix  $H$  with dimensions  $r \times m_c^{d-1}$  and reshapes it into a  $d$  mode tensor where the first mode is of length  $r$  and the remaining  $d - 1$  modes are of length  $m_c$ .

In the scenario where it is possible to access the original tensor  $\mathcal{X}$  twice (we refer to this as the two-pass scenario), we can compute the optimal core  $\mathcal{G}$  given our estimated factor matrices  $Q_i$  to obtain an estimate  $\mathcal{X}_2 = [[\mathcal{G}, Q_1, \dots, Q_d]]$ . See Algorithm 3.10 and Algorithm 3.12 in Section 3.6 for the detailed formulation of the two-pass recovery procedure.

*Remark 3.1.* Note, for a tensor with an exact Tucker decomposition  $\mathcal{X} = [[\mathcal{G}, U_1, \dots, U_d]]$  of rank  $\underbrace{(r, \dots, r)}_d$  the total degrees of freedom in the decomposition is a lower bound on any lossless compression scheme in the absence of any other kind of information about the factorization. For example, the leave-one-out measurements  $B_i$  used to recover a factor matrix  $U_j$  in Algorithm 3.1 which has  $nr - \frac{r(r+1)}{2}$  degrees of freedom, it is necessary in the exact case that the product  $U_j$  and  $G_{[j]} (\Omega_d U_d \otimes \cdots \otimes \Omega_{j+1} U_{j+1} \otimes \Omega_{j-1} U_{j-1} \otimes \cdots \otimes \Omega_1 U_1)$  have column rank at least  $r$  and thus  $m^{d-1} \geq r$ . Similarly we see that  $m_c \geq r$  is a lower bound for the measurements required to recover the  $r^d$  entries of the core. In other words, to recover the rank  $(r, \dots, r)$  core tensor, our sketch along each mode must have dimension at least  $r$ .

With the measurement operators described and the canonical algorithm outlined, we can now state a representative main result. The full unabridged results are found in Section 3.2.5.

**Theorem 3.1** (A Summary Main Result). *Let  $\mathcal{X}$  be a  $d$ -mode tensor of side length  $n$ ,  $\epsilon \in (0, 1)$ ,  $\delta \in (0, \frac{1}{3})$ , and  $r \in [n] := \{1, \dots, n\}$ . Denote an optimal rank  $\mathbf{r} = (r, \dots, r) \in \mathbb{R}^d$  ( $d \geq 2$ ) Tucker approximation of  $\mathcal{X}$  by  $[[\mathcal{X}]]_{\mathbf{r}}^{\text{opt}}$ . There exists a one-pass recovery algorithm (see Algorithm 3.1) that outputs a Tucker factorization  $\mathcal{X}_1 = [[\mathcal{H}, Q_1, \dots, Q_d]]$  of  $\mathcal{X}$  from leave-one-out linear measurements that will satisfy*

$$\|\mathcal{X} - \mathcal{X}_1\|_2 \leq (1 + e^\epsilon) \sqrt{\frac{d(1 + \epsilon)}{1 - \epsilon}} \left\| \mathcal{X} - [[\mathcal{X}]]_{\mathbf{r}}^{\text{opt}} \right\|_2 \quad (3.6)$$

with probability at least  $1 - \delta$ . The total number of linear measurements the algorithm will use is

---

**Algorithm 3.1** One Pass HOSVD Recovery from Leave-One-Out Measurements
 

---

**input :**
 $B_i \in \mathbb{R}^{n \times m^{d-1}}$  for  $i \in [d]$  leave-one-out measurements

 $\Omega_{(i,i)} \in \mathbb{R}^{n \times n}$  for  $i \in [d]$  full-rank sensing matrix for uncompressed mode

 $\mathcal{B}_c$  a  $d$  mode tensor of measurements with side lengths  $m_c$ 
 $\Phi_i \in \mathbb{R}^{m_c \times n}$  for  $i \in [d]$  measurement matrices for core measurements

 $\mathbf{r} = (r, r, \dots, r)$  the rank of the HOSVD

**output :**  $\hat{\mathcal{X}} = [[\mathcal{H}, Q_1, \dots, Q_d]]$ 
**# Factor matrix recovery**
**for**  $i \in [d]$  **do**

 # Solve  $n \times n$  linear system

 Solve  $\Omega_{(i,i)} F_i = B_i$  for  $F_i$ 

 # Compute SVD and keep the  $r$  leading singular vectors

 $U, \Sigma, V^T \leftarrow \text{SVD}(F_i)$ 
 $Q_i \leftarrow U_{:,r}$ 
**end**
**# Core recovery**
**for**  $i \in [d]$  **do**

 # unfold measurements, mode- $i$  fibers are columns, size  $m_c \times r^{(i-1)} m_c^{d-1-(i-1)}$ 
 $H \leftarrow \text{unfold}(\mathcal{B}_c, m_c \times r^{(i-1)} m_c^{d-1-(i-1)}, \text{mode} = i)$ 

 # Undo the mode- $i$  measurement operator and factor's action by finding least square solution to  $m_c \times r$  over-determined linear system

 Solve  $\Phi_i Q_i H_{\text{new}} = H$  for  $H_{\text{new}}$ 

# reshape the flattened partially solved core into a tensor

 $\mathcal{B}_c \leftarrow \text{fold}(H_{\text{new}}, \underbrace{r \times r \cdots \times r}_i \times \underbrace{m_c \times \cdots \times m_c}_{d-i}, \text{mode} = i)$ 

 # Each iteration  $m_c \rightarrow r$  in  $i$ th mode

**end**
 $\mathcal{H} \leftarrow \mathcal{B}_c$ 


---

bounded above by  $m'(r, d, \epsilon, n, \delta) = \left[ nd + \frac{Crd^2}{\epsilon^2} \ln \left( \frac{dn^d}{\delta} \right) \right] \left( \frac{Crd^2}{\epsilon^2} \ln \left( \frac{dn^d}{\delta} \right) \right)^{d-1}$ , where  $C > 0$  is an absolute constant. Furthermore, the recovery algorithm (Algorithm 3.1) runs in time  $o(n^d)$  for large  $n \gg r$ .

*Remark 3.2.* More generally and more exactly, the number of measurements is always  $ndm^{d-1} + m_c^d$  where  $m$  and  $m_c$  are as in Theorem 3.9 for Kronecker structured measurements, or  $d\tilde{m}n + m_c^d$  where  $\tilde{m}$  and  $m_c$  are as in Theorem 3.11 in the case of Khatri-Rao structured measurements.

*Proof.* Combine Theorem 3.9 (or Theorem 3.11) with Theorem 3.10 and Lemma 1.5.  $\square$

Although our bound almost certainly not tight, overall we demonstrate the procedure has errors

that depend on the number of degrees of freedom of the sketched tensor, not its full size; as is typical in similar compressive sensing scenarios. Furthermore, in our numerical experiments we demonstrate some of the main trade-offs involved when choosing between Kronecker or Khatri-Rao constructed sketches in terms of approximation error and performance.

### 3.1.6 Projection Cost Preserving (PCP) Sketches

The following definition and related results are needed to conduct our probabilistic error analysis of the various leave-one-out measurements and Algorithm 3.1. We will rely heavily on some definitions in randomized numerical linear algebra and compressive sensing that were introduced in Chapter 1.

**Definition 3.2 (A  $(\epsilon, c, r)$ -Projection Cost Preserving (PCP) sketch).** Let  $\epsilon, c > 0$ , and  $r \in \mathbb{N}$ . A matrix  $\tilde{X} \in \mathbb{R}^{n \times m}$  is a  $(\epsilon, c, r)$ -PCP sketch of  $X \in \mathbb{R}^{n \times N}$  if for all orthogonal projection matrices  $P \in \mathbb{R}^{n \times n}$  with rank at most  $r$ ,

$$(1 - \epsilon) \|X - PX\|_F^2 \leq \|\tilde{X} - P\tilde{X}\|_F^2 + c \leq (1 + \epsilon) \|X - PX\|_F^2 \quad (3.7)$$

holds.

Note the above property first appeared in this form in [13, Definition 1] (see also, however, [17, Definition 13] for the statement of an equivalent property in a different form that appeared earlier). The next lemma can be used to construct random matrices that are PCP sketches of a given matrix  $X$  with high probability. Before the lemma can be stated, however, we will need one additional definition.

**Definition 3.3 (Head-Tail Split).** For any  $A \in \mathbb{R}^{m \times n}$ , we can split  $A$  into his leading  $r$ -term and its tail  $(n - r)$ -term Singular Value Decomposition (SVD) components. That is, consider the SVD of  $A = U\Sigma V^T$ . For any  $r \leq \text{rank}(A)$ , let  $U_r \in \mathbb{R}^{m \times r}$  and  $V_r \in \mathbb{R}^{n \times r}$  denote the first  $r$  columns of  $U \in \mathbb{R}^{m \times m}$  and  $V \in \mathbb{R}^{n \times n}$ , respectively. We then define  $A_r := U_r U_r^T A = A V_r V_r^T$  to be  $A$ 's best rank  $r$  approximation with respect to  $\|\cdot\|_F$ . Furthermore, we denote the tail term by  $A_{\setminus r} := A - A_r$ .

We will now detail how matrices with the OSE and AMM (recall Definitions 1.23 and 1.25) properties interacting with matrices derived from  $X \in \mathbb{R}^{n \times N}$  will yield PCP sketches of  $X$  with

high probability. Variants of the following result are proven in [11, 53]. We include the proof in Section 3.5.2 for the sake of completeness.

**Theorem 3.2** (Projection-Cost-Preservation via the AMM and OSE properties). *Let  $X \in \mathbb{R}^{n \times N}$  of rank  $\tilde{r} \leq \min\{n, N\}$  have the full SVD  $X = U\Sigma V^T$ , and let  $V_{r'} \in \mathbb{R}^{N \times r'}$  denote the first  $r'$  columns of  $V \in \mathbb{R}^{N \times N}$  for all  $r' \in [N]$ . Fix  $r \in [n]$  and consider the head-tail split  $X = X_r + X_{\setminus r}$ . If  $\Omega \in \mathbb{R}^{m \times N}$  satisfies*

1. *subspace embedding property (1.10) with  $\epsilon \leftarrow \frac{\epsilon}{3}$  for  $A \leftarrow X_r^T$ ,*
2. *approximate multiplication property (1.13) with  $\epsilon \leftarrow \frac{\epsilon}{6\sqrt{\min\{r, \tilde{r}\}}}$  for  $A \leftarrow X_{\setminus r}$  and  $B \leftarrow V_{\min\{r, \tilde{r}\}}$ ,*
3. *JL property (1.8) with  $\epsilon \leftarrow \frac{\epsilon}{6}$  for  $S \leftarrow \{\text{the } n \text{ columns of } X_{\setminus r}^T\}$ , and*
4. *approximate multiplication property (1.13) with  $\epsilon \leftarrow \frac{\epsilon}{6\sqrt{r}}$  for  $A \leftarrow X_{\setminus r}$  and  $B \leftarrow X_{\setminus r}^T$ ,*

*then  $\tilde{X} := X\Omega^T$  is an  $(\epsilon, 0, r)$ -PCP sketch of  $X$ .*

*Proof.* See Section 3.5.2. □

The following lemma can be used to construct PCP sketches from random matrices with the JL property.

**Lemma 3.1** (The JL property provides PCP sketches). *Let  $X \in \mathbb{R}^{n \times N}$  have rank  $\tilde{r} \leq \min\{n, N\}$ . Fix  $r \in [n]$ . There exist finite sets  $S_1, S_2 \subset \mathbb{R}^N$  (determined entirely by  $X$ ) with cardinalities  $|S_1| \leq \left(\frac{141}{\epsilon}\right)^{\min\{r, \tilde{r}\}}$  and  $|S_2| \leq 16n^2 + n$  such that the following holds: If a random matrix  $\Omega \in \mathbb{R}^{m \times N}$  has both the  $\left(\frac{\epsilon}{6}, \frac{\delta}{2}, \left(\frac{141}{\epsilon}\right)^r\right)$ -JL property for  $S_1$  and the  $\left(\frac{\epsilon}{6\sqrt{r}}, \frac{\delta}{2}, 16n^2 + n\right)$ -JL property for  $S_2$ , then  $X\Omega^T$  will be an  $(\epsilon, 0, r)$ -PCP sketch of  $X$  with probability at least  $1 - \delta$ .*

*Proof.* To ensure property 1 of Theorem 3.2 we can appeal to Lemma 1.6 to see that  $\Omega$  being an  $(\epsilon/6)$ -JL-embedding of a minimal  $\left(\frac{\epsilon}{48}\right)$ -cover of the at most  $\min\{r, \tilde{r}\}$ -dimensional unit ball in the column space of  $X_r^T$  will suffice. Letting  $S_1$  be this aforementioned  $\left(\frac{\epsilon}{48}\right)$ -cover, we can further see that  $|S_1| \leq (141/\epsilon)^{\min\{r, \tilde{r}\}}$  by the proof of Corollary 1.1. Hence, if  $\Omega \in \mathbb{R}^{m \times N}$  has the  $\left(\frac{\epsilon}{6}, \frac{\delta}{2}, \left(\frac{141}{\epsilon}\right)^r\right)$ -JL property for  $S_1$ , then property 1 of Theorem 3.2 will be satisfied with with probability at least  $1 - \frac{\delta}{2}$ .

Applying Lemma 1.8 one can see that there exist sets  $S'_2, S''_2 \subset \mathbb{R}^N$  with  $|S'_2| \leq 2(n +$

$\min\{r, \tilde{r}\}^2 \leq 8n^2$  and  $|S_2''| \leq 2(n+n)^2 = 8n^2$  such that an  $(\epsilon/6\sqrt{r})$ -JL-embedding of  $S_2' \cup S_2''$  will satisfy both properties 2 and 4 of Theorem 3.2. Hence, since  $r \geq 1$ , we can see that an  $(\epsilon/6\sqrt{r})$ -JL-embedding of  $S_2 := S_2' \cup S_2'' \cup S$  will satisfy Theorem 3.2 properties 2 – 4, where  $S$  is defined as per property 3. Noting that  $|S_2| \leq |S_2'| + |S_2''| + |S| \leq 16n^2 + n$ , we can now see that  $\Omega$  will satisfy all of Theorem 3.2's properties 2 – 4 with probability at least  $1 - \frac{\delta}{2}$  if it has the  $\left(\frac{\epsilon}{6\sqrt{r}}, \frac{\delta}{2}, 16n^2 + n\right)$ -JL property for  $S_2$ .

Concluding, the prior two paragraphs in combination with the union bound imply that all of Theorem 3.2's properties 1 – 4 will hold with probability at least  $1 - \delta$  if  $\Omega$  has both the  $\left(\frac{\epsilon}{6}, \frac{\delta}{2}, \left(\frac{141}{\epsilon}\right)^r\right)$ -JL property for  $S_1$  and the  $\left(\frac{\epsilon}{6\sqrt{r}}, \frac{\delta}{2}, 16n^2 + n\right)$ -JL property for  $S_2$ . An application of Theorem 3.2 now finishes the proof.  $\square$

Using Lemma 3.1 one can now prove the following corollary of Theorem 1.1 which demonstrates the existence of a PCP sketch for any fixed matrix  $X$ .

**Corollary 3.1.** *Fix  $X \in \mathbb{R}^{n \times N}$  and  $r \in [n]$ . Let  $\Omega \in \mathbb{R}^{m \times N}$  be a random matrix with mean zero, variance  $\frac{1}{m}$ , independent sub-Gaussian entries. Then,  $X\Omega^T$  will be an  $(\epsilon, 0, r)$ -PCP sketch of  $X$  with probability at least  $1 - \delta$  provided that*

$$m \geq C \frac{r}{\epsilon^2} \max \left\{ \ln \left( \frac{C_1}{\epsilon \delta} \right), \ln \left( \frac{C_2 n}{\delta} \right) \right\},$$

where  $C_1, C_2 > 0$  are absolute constants, and  $C > 0$  is an absolute constant that only depends on the sub-Gaussian norms (see Definition 1.22) of  $\Omega$ 's entries.

*Proof.* Apply Theorem 1.1 to the finite set  $S_1$  guaranteed by Lemma 3.1 with  $\epsilon \leftarrow \frac{\epsilon}{6}$  and  $\delta \leftarrow \frac{\delta}{2}$ . Similarly, apply Theorem 1.1 to the finite set  $S_2$  guaranteed by Lemma 3.1 with  $\epsilon \leftarrow \frac{\epsilon}{6\sqrt{r}}$  and  $\delta \leftarrow \frac{\delta}{2}$ . The result now follows by Lemma 3.1.  $\square$

We finish here by noting that Corollary 3.1 is just one example of a PCP sketching result that one can prove with relative ease using Lemma 3.1. Indeed, Lemma 3.1 can be combined with other standard results concerning more structured matrices with the JL property (see, e.g., [38, 30, 3, 27]) to produce similar theorems where  $\Omega$  has a fast matrix-vector multiply.

### 3.2 Proofs of Our Main Results on Leave-One-Measurements and Recovery

The objective is to show that we can retrieve an accurate low rank Tucker approximation of a tensor  $\mathcal{X}$  via Algorithm 3.1 from valid sets of leave-one-out and core measurements as described in Section 3.1. We will denote the approximation of  $\mathcal{X}$  output by Algorithm 3.1 as  $\mathcal{X}_1$  here to emphasize that a single pass over the original data tensor  $\mathcal{X}$  suffices in order to compute the linear input measurements required by Algorithm 3.1. Hence, Algorithm 3.1 in this setting is an example of a *streaming algorithm* which doesn't need to store a copy the original uncompressed tensor  $\mathcal{X}$  in memory in order to successfully approximate it. Nonetheless, we wish to show that this algorithm still produces a quasi-optimal approximation of  $\mathcal{X}$  in the sense of (1.7) with high probability when given such highly compressed linear input measurements. Particular choices for measurement ensembles will make explicit the dependence on other parameters of the problem (these choices define specializations of Algorithm 3.1, and are summarized as Algorithm 3.9 and 3.11 in Section 3.6).

More specifically, in this section we will show that with high probability for a given  $d$ -mode tensor  $\mathcal{X}$ , error tolerance  $\epsilon > 0$ , and chosen rank truncation parameter  $r$ , that

$$\|\mathcal{X} - \mathcal{X}_1\|_2 \leq (1 + e^\epsilon) \sqrt{\frac{1 + \epsilon}{1 - \epsilon} \sum_{j=1}^d \Delta_{r,j}} \quad (3.8)$$

will hold whenever Algorithm 3.1 is provided with sufficiently informative input measurements. Here the  $\Delta_{r,j}$  are defined as per Lemma 1.5, and “sufficiently informative” means that (i) the leave-one-out measurements used to form  $\mathcal{X}_1$  are of sufficient size to satisfy several PCP properties, and that (ii) the core measurements used to form  $\mathcal{X}_1$  are of sufficient size to ensure the accurate solution of least squares problems computed as part of Algorithm 3.1. Finally, we note that one can see from (3.8) together with Lemma 1.5 that Algorithm 3.1 will perfectly recover exactly low Tucker-rank tensors if the rank parameter  $r$  is made sufficiently large.

In order to prove that (3.8) holds, we will need to also consider a weaker variant of Algorithm 3.1 which permits a *second* pass of the data tensor  $\mathcal{X}$ . These weaker algorithms will first compute estimates of the factors of the tensor  $Q_i$  as Algorithm 3.1 does, but thereafter will be allowed to use

those factors to operate on the original tensor  $\mathcal{X}$  in order to approximate its core (see Algorithms 3.10 and 3.12). We denote the estimate of the tensor that results from this procedure as  $\mathcal{X}_2$  to emphasize that it requires a second accesses to the original data during core recovery. Note that such two-pass algorithms are of less practical value in the big data and compressive sensing settings since it is often not possible to directly access the data tensor again after the initial compressed measurements have been taken in these scenarios. Nevertheless, this two-pass estimate will be extremely useful when proving (3.8). In particular, our proof will result from the following triangle inequality:

$$\|\mathcal{X} - \mathcal{X}_1\|_2 = \|\mathcal{X} - \mathcal{X}_1 + \mathcal{X}_2 - \mathcal{X}_2\|_2 \leq \underbrace{\|\mathcal{X} - \mathcal{X}_2\|_2}_{\text{Term I}} + \underbrace{\|\mathcal{X}_1 - \mathcal{X}_2\|_2}_{\text{Term II}}. \quad (3.9)$$

Bounding Term I in (3.9) will be the subject of Section 3.2.1. As we shall see, bounding Term I is straightforward if we have that leave-one-out measurements imply various PCP properties; and so the main work of Sections 3.2.2 and 3.2.3 will be to demonstrate how, for the structured choices of measurement operators considered herein, we can ensure that the PCP property is satisfied. Bounding Term II, on the other hand, will require us to apply a bound on the error incurred by solving sketched least square problems on a carefully partitioned re-expression of the Term II error. That argument is the subject of Section 3.2.4. Finally, we combine our analysis of these two error terms along with particular choices for measurement operators to state the full versions of our main results in Section 3.2.5.

### 3.2.1 Bounding $\|\mathcal{X} - \mathcal{X}_2\|_2$

In the two pass scenario, we first compute estimates for the factor matrices,  $Q_i$  (see Algorithm 3.5), using leave-one-out measurements  $B_i$  for each  $i \in [d]$ . Then, using these factor matrices, we wish to solve

$$\arg \min_{\mathcal{H}} \|\mathcal{X} - [[\mathcal{H}, Q_1, Q_2, \dots, Q_d]]\|_2.$$

One can see that the solution will be

$$\mathcal{G} := \mathcal{X} \times_1 Q_1^T \times_2 Q_2^T \cdots \times_d Q_d^T$$

(see, e.g., [16]). Let

$$\mathcal{X}_2 := \mathcal{G} \times_1 Q_1 \times_2 Q_2 \cdots \times_d Q_d \quad (3.10)$$

denote the estimate obtained from a two-pass recovery procedure (i.e., Algorithm 3.10 or 3.12).

Additionally, we note the following fact about modewise products (see, e.g., [67, Lemma 1]):

$$\mathcal{X} \times_i A \times_i B = \mathcal{X} \times_i (BA).$$

As a result, if we are permitted a second pass over  $\mathcal{X}$  to compute the core we have that

$$\begin{aligned} \mathcal{X}_2 &= \mathcal{G} \times_1 Q_1 \times_2 Q_2 \cdots \times_d Q_d \\ &= \left( \mathcal{X} \times_1 Q_1^T \times_2 Q_2^T \cdots \times_d Q_d^T \right) \times_1 Q_1 \times_2 Q_2 \cdots \times_d Q_d \\ &= \mathcal{X} \times_1 Q_1 Q_1^T \times_2 \cdots \times_d Q_d Q_d^T \end{aligned}$$

Using this expression we can now bound the two pass error term  $\|\mathcal{X} - \mathcal{X}_2\|_2$ .

**Theorem 3.3** (Error bound for Two-Pass  $\|\mathcal{X} - \mathcal{X}_2\|_2$ ). *Suppose  $\tilde{X}_{[j]} := X_{[j]} \Omega_{-j}^T \in \mathbb{R}^{n \times m^{d-1}}$  are  $(\epsilon, 0, r)$ -PCP sketches of  $X_{[j]}$  for each  $j \in [d]$ . If  $Q_j \in \mathbb{R}^{n \times r}$  for  $r \leq m$  are factor matrices obtained from Algorithm 3.5, then*

$$\|\mathcal{X} - \mathcal{X}_2\|_2 = \|\mathcal{X} - \mathcal{X} \times_1 Q_1 Q_1^T \times_2 \cdots \times_d Q_d Q_d^T\|_2 \leq \sqrt{\frac{1+\epsilon}{1-\epsilon} \sum_{j=1}^d \Delta_{r,j}}. \quad (3.11)$$

*Proof.* Since the  $Q_i Q_i^T$  are orthogonal projectors, we have by [62, Theorem 5.1] that

$$\|\mathcal{X} - \mathcal{X} \times_1 Q_1 Q_1^T \times_2 \cdots \times_d Q_d Q_d^T\|_2^2 \leq \sum_{j=1}^d \left\| \mathcal{X} - \mathcal{X} \times_j Q_j Q_j^T \right\|_2^2. \quad (3.12)$$

From Algorithm 3.5 (recalling that  $\Omega_{(j,j)}^{-1} B_j = X_{[j]} \Omega_{-j}^T = \tilde{X}_{[j]}$ ), we have that the  $Q_j$ 's are the best rank- $r$  approximations for their respective sketched problems, since

$$Q_j = \arg \min_{\substack{\text{rank}(Q)=r \\ Q^T Q = I_r}} \left\| \tilde{X}_{[j]} - Q Q^T \tilde{X}_{[j]} \right\|_F$$

by the Eckart–Young Theorem.

Now suppose that each  $U_j \in \mathbb{R}^{n \times r}$  forms an optimal rank  $r$  approximation to  $X_{[j]}$  in the sense that

$$U_j = \arg \min_{\substack{\text{rank}(U)=r \\ U^T U = I_r}} \|X_{[j]} - U U^T X_{[j]}\|_F.$$

By the hypothesis that  $\tilde{X}_{[j]}$  is a  $(\epsilon, 0, r)$ -PCP sketch of  $X_{[j]}$ , we have that

$$\begin{aligned} (1 - \epsilon) \left\| X - X \times_j Q_j Q_j^T \right\|_2^2 &= (1 - \epsilon) \left\| X_{[j]} - Q_j Q_j^T X_{[j]} \right\|_F^2 \\ &\leq \left\| \tilde{X}_{[j]} - Q_j Q_j^T \tilde{X}_{[j]} \right\|_F^2 \\ &\leq \left\| \tilde{X}_{[j]} - U_j U_j^T \tilde{X}_{[j]} \right\|_F^2 \\ &\leq (1 + \epsilon) \left\| X_{[j]} - U_j U_j^T X_{[j]} \right\|_F^2 = (1 + \epsilon) \Delta_{r,j}, \end{aligned}$$

where we have used the definition of  $(\epsilon, 0, r)$ -PCP sketches in the first and third inequalities. After a rearrangement of terms, substituting the above into (3.12) now yields the inequality in (3.11).  $\square$

We have now established in Theorem 3.3 that we have a quasi-optimal error bound for Term I in (3.9) whenever our leave-one-out measurement matrices  $\Omega_{-j}^T$  yield  $(\epsilon, 0, r)$ -PCP sketches of all  $d$  unfoldings  $X_j$ . Next, we will demonstrate how to ensure that Kronecker structured and Khatri-Rao structured leave-one-out measurement matrices provide PCP sketches.

### 3.2.2 PCP Sketches via Kronecker-Structured Leave-one-out Measurement Matrices

In this section we study when Kronecker-structured measurement matrices will provide the PCP property. To begin we will show that the JL and OSE properties are inherited under matrix direct sums and compositions. These are useful facts because our overall leave-one-out matrices can be constructed using these operations. In particular, we will follow the example in the last paragraph of Section 3.1.1 and consider a matrix  $\Omega_{-j} \in \mathbb{R}^{m^{d-1} \times n^{d-1}}$  defined as

$$\Omega_{-j} = \bigotimes_{\substack{i=1 \\ i \neq j}}^d \Omega_i = \prod_{i'=1}^{d-1} \tilde{\Omega}_{i'} \quad \text{for } \Omega_i \in \mathbb{R}^{m \times n} \quad (3.13)$$

where

$$\tilde{\Omega}_{i'} := \underbrace{I_n \otimes \cdots \otimes I_n}_{d-1-i'} \otimes \Omega_{i_j(i')} \otimes \underbrace{I_m \otimes \cdots \otimes I_m}_{i'-1} \in \mathbb{R}^{m^{i'} n^{d-1-i'} \times m^{i'-1} n^{d-i'}} \quad (3.14)$$

for

$$i_j(i') := \begin{cases} i' & \text{if } i' < j \\ i' + 1 & \text{if } i' \geq j \end{cases}.$$

Here,  $I_n$  denotes an  $n \times n$  identity matrix.

The next three lemmas will be used to help show that  $\Omega_{-j}$  as defined in (3.13) inherits both the JL and OSE properties from its component  $\Omega_i$  matrices. Having established this, we can then use, e.g., Lemma 3.1 to prove PCP sketching results for such Kronecker-structured  $\Omega_{-j}$ .

**Lemma 3.2.** *Suppose that  $\Omega_1 \in \mathbb{R}^{m_1 \times N_1}$  and  $\Omega_2 \in \mathbb{R}^{m_2 \times N_2}$  are two random matrices. Denote their matrix direct sum by  $\Omega = \Omega_1 \oplus \Omega_2 \in \mathbb{R}^{(m_1+m_2) \times (N_1+N_2)}$ . Then,*

1. *If  $\Omega_1$  and  $\Omega_2$  are  $(\epsilon, \delta_1, p)$  and  $(\epsilon, \delta_2, p)$ -JLs respectively, then  $\Omega$  is an  $(\epsilon, \delta_1 + \delta_2, p)$ -JL.*

2. *If  $\Omega_1$  and  $\Omega_2$  are  $(\epsilon, \delta_1, r)$  and  $(\epsilon, \delta_2, r)$ -OSEs respectively, then  $\Omega$  is an  $(\epsilon, \delta_1 + \delta_2, r)$ -OSE.*

*Proof. Part 1.:* Consider a set  $S \subset \mathbb{R}^{N_1+N_2}$  with cardinality  $p$ . Let  $\mathbf{z} \in S$ . Group the first  $N_1$  coordinates of  $\mathbf{z}$  into  $\mathbf{x} \in \mathbb{R}^{N_1}$  and the last  $N_2$  coordinates of  $\mathbf{z}$  into  $\mathbf{y} \in \mathbb{R}^{N_2}$ . Observe that

$$\|\Omega \mathbf{z}\|_2^2 = \left\| \begin{bmatrix} \Omega_1 & 0 \\ 0 & \Omega_2 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \right\|_2^2 = \|\Omega_1 \mathbf{x}\|_2^2 + \|\Omega_2 \mathbf{y}\|_2^2 \leq (1 + \epsilon) \left( \|\mathbf{x}\|_2^2 + \|\mathbf{y}\|_2^2 \right) = (1 + \epsilon) \|\mathbf{z}\|_2^2$$

will hold whenever both  $\|\Omega_1 \mathbf{x}\|_2^2 \leq (1 + \epsilon) \|\mathbf{x}\|_2^2$  and  $\|\Omega_2 \mathbf{y}\|_2^2 \leq (1 + \epsilon) \|\mathbf{y}\|_2^2$  hold. The  $(1 - \epsilon)$ -distortion lower bound is similar. As a result, we can use the union bound to see that  $\Omega$  will have the  $(\epsilon, \delta_1 + \delta_2, p)$ -JL property.

*Part 2.:* Suppose  $X \in \mathbb{R}^{n \times (N_1+N_2)}$  has rank  $r$ . Let  $X_1$  and  $X_2$  denote the sub-matrices of  $X$  containing the first  $N_1$  and last  $N_2$  columns of  $X$ , respectively. Note that both  $X_1$  and  $X_2$  have at most rank  $r$ . Furthermore, note also that

$$\|\Omega X^T \mathbf{y}\|_2^2 = \left\| \begin{pmatrix} \Omega_1 X_1^T \\ \Omega_2 X_2^T \end{pmatrix} \mathbf{y} \right\|_2^2 \leq (1 + \epsilon) \left( \|X_1^T \mathbf{y}\|_2^2 + \|X_2^T \mathbf{y}\|_2^2 \right) = (1 + \epsilon) \|X^T \mathbf{y}\|_2^2.$$

will hold for any arbitrary vector  $\mathbf{y} \in \mathbb{R}^n$  whenever both  $\|\Omega_1 X_1^T \mathbf{y}\|_2^2 \leq (1 + \epsilon) \|X_1^T \mathbf{y}\|_2^2$  and  $\|\Omega_2 X_2^T \mathbf{y}\|_2^2 \leq (1 + \epsilon) \|X_2^T \mathbf{y}\|_2^2$  hold. The  $(1 - \epsilon)$ -distortion lower bound is similar. As a result, we can see that  $\Omega$  will be an  $(\epsilon, \delta_1 + \delta_2, r)$ -OSE by the union bound.  $\square$

Note that there is no requirement that  $\Omega_1$  and  $\Omega_2$  need to be independent in Lemma 3.2. This is crucial for the next lemma, which will involve many copies of the same measurement matrix.

**Lemma 3.3** (Direct Sums Inherit the OSE and JL Properties). *For some  $i' \in [d - 1]$ , let  $\tilde{\Omega}_{i'}$  be defined as in (3.14) and set  $\Omega := \Omega_{i_j(i')}$   $\in \mathbb{R}^{m \times n}$ .*

1. *If  $\Omega$  has the  $(\epsilon, \frac{\delta}{m^{i'-1}n^{d-i'-1}}, r)$ -OSE property, then  $\tilde{\Omega}_{i'}$  will have the  $(\epsilon, \delta, r)$ -OSE property.*
2. *If  $\Omega$  has the  $(\epsilon, \frac{\delta}{m^{i'-1}n^{d-i'-1}}, p)$ -JL property, then  $\tilde{\Omega}_{i'}$  will have the  $(\epsilon, \delta, p)$ -JL property.*

*Proof.* First consider the rearrangement of  $\tilde{\Omega}_{i'}$  in (3.14) defined as follows

$$\tilde{\Omega} := \underbrace{I_m \otimes I_m \otimes \cdots \otimes I_m}_{i'-1} \otimes \underbrace{I_n \otimes I_n \otimes \cdots \otimes I_n}_{d-1-i'} \otimes \Omega.$$

Note that the Kronecker product of two identity matrices is itself an identity matrix. Thus, we can rewrite this as simply

$$\tilde{\Omega} = \underbrace{I_m \otimes I_m \otimes \cdots \otimes I_m}_{i'-1} \otimes \underbrace{I_n \otimes I_n \otimes \cdots \otimes I_n}_{d-1-i'} \otimes \Omega = \begin{bmatrix} \Omega & 0 & \cdots & 0 \\ 0 & \Omega & \cdots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \cdots & \Omega \end{bmatrix}.$$

That is, we have a block diagonal matrix with  $\bar{m} = m^{i'-1}n^{d-i'-1}$  copies of  $\Omega$  along its diagonal. Thus, if  $\Omega$  has either the  $(\epsilon, \delta/\bar{m}, r)$ -OSE or the  $(\epsilon, \delta/\bar{m}, p)$ -JL property, repeated applications of Lemma 3.2 will then establish the desired OSE or JL property for  $\tilde{\Omega}$ .

Now consider  $\tilde{\Omega}_{i'}$  as in (3.14). There exist unitary (permutation) matrices  $L$  and  $R$  which interchange rows and columns such that

$$\begin{aligned} L\tilde{\Omega}R &= L \left( \underbrace{I_m \otimes I_m \otimes \cdots \otimes I_m}_{i'-1} \otimes \underbrace{I_n \otimes I_n \otimes \cdots \otimes I_n}_{d-1-i'} \otimes \Omega \right) R \\ &= L \left( I_{m^{i'-1}n^{d-1-i'}} \otimes \Omega \right) R = \underbrace{I_n \otimes I_n \otimes \cdots \otimes I_n}_{d-1-i'} \otimes \Omega \otimes \underbrace{I_m \otimes I_m \otimes \cdots \otimes I_m}_{i'-1} \\ &= I_{n^{d-1-i'}} \otimes \Omega \otimes I_{m^{i'-1}} = \tilde{\Omega}_{i'}. \end{aligned}$$

Noting that both the OSE and JL properties are invariant to unitary transformations of a given random matrix, one can now see that  $\tilde{\Omega}_{i'} = L\tilde{\Omega}R$  will indeed have the same desired OSE or JL property as was established for  $\tilde{\Omega}$ .  $\square$

Lemma 3.3 allows us to infer JL and OSE properties of the  $\tilde{\Omega}_{i'}$  matrices in (3.14) from the properties of the smaller random matrices  $\Omega_i \in \mathbb{R}^{m \times n}$  appearing in (3.13). The next lemma will allow us to then use these inferred properties of the  $\tilde{\Omega}_{i'}$  matrices to derive OSE and JL properties for  $\Omega_{-j}$  from (3.13) in terms of the properties of its component  $\Omega_i \in \mathbb{R}^{m \times n}$ .

**Lemma 3.4** (A Composition Lemma for the OSE and JL Properties). *Let  $\epsilon \in (0, 1)$  and  $\tilde{\Omega}_{i'} \in \mathbb{R}^{m^{i'} n^{d-1-i'} \times m^{i'-1} n^{d-i'}}$  for  $i' \in [d-1]$ .*

1. *If  $\tilde{\Omega}_{i'}$  is an  $\left(\frac{\epsilon}{2(d-1)}, \frac{\delta}{d-1}, r\right)$ -OSE for all  $i' \in [d-1]$ , then  $\tilde{\Omega} = \prod_{i'=1}^{d-1} \tilde{\Omega}_{i'}$  is an  $(\epsilon, \delta, r)$ -OSE.*
2. *If  $\tilde{\Omega}_{i'}$  is an  $\left(\frac{\epsilon}{2(d-1)}, \frac{\delta}{d-1}, p\right)$ -JL for all  $i' \in [d-1]$ , then  $\tilde{\Omega} = \prod_{i'=1}^{d-1} \tilde{\Omega}_{i'}$  is an  $(\epsilon, \delta, p)$ -JL.*

*Proof. Part 1.:* Let  $Y^T \in \mathbb{R}^{n^{d-1} \times n}$  be an arbitrary matrix of rank at most  $r$ . Denote  $\tilde{Y}_{i'} = (\tilde{\Omega}_{i'} \dots \tilde{\Omega}_1) Y^T \in \mathbb{R}^{m^{i'} n^{d-1-i'} \times n}$  for  $i' \in [d-1]$ . Note that each  $\tilde{Y}_{i'}$  has rank at most  $r$ . Fix some  $\mathbf{z} \in \mathbb{R}^n$ . Suppose for the moment that (1.10) holds for each  $i' \in [d-1]$  with  $\Omega = \tilde{\Omega}_{i'}$ ,  $A = \tilde{Y}_{i'-1}$ , and  $\mathbf{x} = \mathbf{z}$ , we have that

$$\begin{aligned}
\|\tilde{\Omega} Y^T \mathbf{z}\|_2^2 &= \|\tilde{\Omega}_{d-1} (\tilde{\Omega}_{d-2} \dots \tilde{\Omega}_1) Y^T \mathbf{z}\|_2^2 \\
&= \|\tilde{\Omega}_{d-1} \tilde{Y}_{d-2} \mathbf{z}\|_2^2 \\
&\leq \left(1 + \frac{\epsilon}{2(d-1)}\right) \|\tilde{Y}_{d-2} \mathbf{z}\|_2^2 \\
&= \left(1 + \frac{\epsilon}{2(d-1)}\right) \|\tilde{\Omega}_{d-2} \tilde{Y}_{d-3} \mathbf{z}\|_2^2 \\
&\vdots \\
&\leq \left(1 + \frac{\epsilon}{2(d-1)}\right)^{d-1} \|Y^T \mathbf{z}\|_2^2 \\
&\leq \left(\frac{1}{1 - \epsilon/2}\right) \|Y^T \mathbf{z}\|_2^2 \\
&\leq (1 + \epsilon) \|Y^T \mathbf{z}\|_2^2,
\end{aligned}$$

where we have used the general bound  $(1 + k/n)^n \leq e^k \leq (1 - k)^{-1}$  for  $k \in [0, 1)$  in the second to

last inequality. Similarly, for a lower bound one can see that

$$\begin{aligned}\|\tilde{\Omega}Y^T\mathbf{z}\|_2^2 &\geq \left(1 - \frac{\epsilon}{2(d-1)}\right)^{d-1} \|Y^T\mathbf{z}\|_2^2 \\ &\geq (1 - \epsilon) \|Y^T\mathbf{z}\|_2^2.\end{aligned}$$

Union bounding over the failure probability that (1.10) holds for each  $i' \in [d-1]$  as supposed above now yields the desired result.

*Part 2.:* An essentially identical arguments also applies to obtain the desired JL property result.  $\square$

We now have all the necessary results to show how the component maps  $\Omega_i$  of a Kronecker structured measurement ensemble as per (3.13) can guarantee a Kronecker sketch with the projection cost preserving property.

**Theorem 3.4** (Kronecker Products of JL matrices yield PCP Sketchs). *Let  $\epsilon \in (0, 1)$ ,  $X \in \mathbb{R}^{n \times n^{d-1}}$  have rank  $r \in [n]$ , and  $\Omega_{-j} \in \mathbb{R}^{m^{d-1} \times n^{d-1}}$  be defined as in (3.13) and (3.14). Furthermore, suppose that the  $\Omega_i \in \mathbb{R}^{m \times n}$  in (3.13) have both the*

1.  $\left(\frac{\epsilon}{12(d-1)}, \frac{\delta}{2(d-1)n^{d-2}}, \left(\frac{141}{\epsilon}\right)^r\right)$ -JL property, and the
2.  $\left(\frac{\epsilon}{12\sqrt{r}(d-1)}, \frac{\delta}{2(d-1)n^{d-2}}, 16n^2 + n\right)$ -JL property

for all  $i' \in [d-1]$ . Then,  $X\Omega_{-j}^T$  will be an  $(\epsilon, 0, r)$ -PCP sketch of  $X$  with probability at least  $1 - \delta$ .

*Proof.* By Lemma 3.1 we know that  $X\Omega_{-j}^T$  will be an  $(\epsilon, 0, r)$ -PCP sketch of  $X$  with probability at least  $1 - \delta$  if  $\Omega_{-j}$  has both the  $\left(\frac{\epsilon}{6}, \frac{\delta}{2}, \left(\frac{141}{\epsilon}\right)^r\right)$ -JL property and the  $\left(\frac{\epsilon}{6\sqrt{r}}, \frac{\delta}{2}, 16n^2 + n\right)$ -JL property.

In fact, by Lemma 3.4 we can further see that it suffices to have the  $\tilde{\Omega}_{i'}$  from (3.13) and (3.14) have both the

1.  $\left(\frac{\epsilon}{12(d-1)}, \frac{\delta}{2(d-1)}, \left(\frac{141}{\epsilon}\right)^r\right)$ -JL property, and the
2.  $\left(\frac{\epsilon}{12\sqrt{r}(d-1)}, \frac{\delta}{2(d-1)}, 16n^2 + n\right)$ -JL property

for all  $i' \in [d-1]$ . Finally, looking now at Lemma 3.3 for each  $i' \in [d-1]$  we can see that the assumed properties of the  $\Omega_i \in \mathbb{R}^{m \times n}$  in (3.13) will guarantee both of these sufficient conditions.  $\square$

The following corollary of Theorem 3.4 guarantees a Kronecker sketch with the projection cost preserving property when the component matrices  $\Omega_i$  in (3.13) are sub-Gaussian random matrices.

**Corollary 3.2** (Kronecker Products of sub-Gaussian Matrices Yield PCP Sketches). *Suppose  $\mathcal{X}$  is a real valued  $d$ -mode tensor with side-lengths all equal to  $n$ . Let  $\epsilon \in (0, 1), \delta \in (0, 1), r \in [n], j \in [d]$ . If  $\Omega_{-j} = \bigotimes_{\substack{i=1 \\ i \neq j}}^d \Omega_i \in \mathbb{R}^{m^{d-1} \times n^{d-1}}$  defined as in (3.13) with random matrices  $\Omega_i \in \mathbb{R}^{m \times n}$  having i.i.d centered variance  $m^{-1}$ , sub-Gaussian entries such that*

$$m \geq \max \left\{ \frac{C_1 r (d-1)^2}{\epsilon^2} \ln \left( \frac{n^d (d-1)}{\delta} \right), \frac{C_2 (d-1)^2}{\epsilon^2} \ln \left( \left( \frac{141}{\epsilon} \right)^r \frac{n^{d-2} (d-1)}{\delta} \right) \right\}$$

for absolute constants  $C_1, C_2 > 0$  then the sketched unfolding  $\tilde{X}_{[j]} = X_{[j]} \Omega_{-j}^T \in \mathbb{R}^{n \times m^{d-1}}$  is an  $(\epsilon, 0, r)$ -PCP sketch of  $X_{[j]}$  with probability at least  $1 - \delta$ .

*Proof.* To obtain the first quantity maximized over we apply Theorem 1.1 with  $\epsilon \leftarrow \frac{\epsilon}{12\sqrt{r}(d-1)}$ ,  $\delta \leftarrow \frac{\delta}{2(d-1)n^{d-2}}$ , and  $|S| \leftarrow 16n^2 + n$ . Similarly, for the second quantity maximized over we apply Theorem 1.1 with  $\epsilon \leftarrow \frac{\epsilon}{12(d-1)}$ ,  $\delta \leftarrow \frac{\delta}{2(d-1)n^{d-2}}$ , and  $|S| \leftarrow \left( \frac{141}{\epsilon} \right)^r$ . The result now follows from Theorem 3.4.  $\square$

*Remark 3.3.* Note that Theorem 3.4 is quite general, requiring only that the random matrices  $\Omega_i$  in (3.13) should be drawn from some distribution having a couple JL properties. As a result of this generality, its Corollary 3.2 concerning sub-Gaussian component matrices turns out to be sub-optimal by (at least a) factor of  $d$  in that setting. To obtain a slightly sharper result in  $d$  for sub-Gaussian  $\Omega_i$  we recommend replacing our implicit use of Lemma 3.3 in the proof of Corollary 3.2 (via Theorem 3.4) with [33, Lemma 14] instead.

### 3.2.3 PCP Sketches via Khatri-Rao Structured Leave-one-out Measurement Matrices

In this section we study how to ensure that Khatri-Rao structured leave-one-out measurement matrices will provide the PCP property. To start we will first show that random Khatri-Rao structured measurement maps, denoted in this section by

$$\Omega_{-j} = \frac{1}{m} \Omega_1 \cdot \Omega_2 \cdot \dots \cdot \Omega_{j-1} \cdot \Omega_{j+1} \cdot \Omega_d \text{ where } \Omega_i \in \mathbb{R}^{m \times n} \forall i \in [d] \setminus \{j\},$$

will have the JL property whenever all their component matrices  $\Omega_i$  have i.i.d. sub-Gaussian entries. Having established this, we can then use, e.g., Lemma 3.1 to prove PCP sketching results for such Khatri-Rao Structured  $\Omega_{-j}$ .

**Theorem 3.5.** *Let  $\epsilon > 0, 0 < \delta \leq e^{-2}$ , and  $\Omega_{-j} = \frac{1}{m}\Omega_1 \cdot \Omega_2 \cdot \dots \cdot \Omega_{j-1} \cdot \Omega_{j+1} \cdot \Omega_d$  where all the  $\Omega_i \in \mathbb{R}^{m \times n}$  for  $i \in [d] \setminus \{j\}$  have i.i.d. mean zero, variance one, sub-Gaussian entries. Then  $\Omega_{-j}$  is an  $(\epsilon, \delta, k)$ -JL whenever*

$$m \geq C^{d-1} \max \left\{ \epsilon^{-2} \log \frac{k}{\delta}, \epsilon^{-1} \left( \log \frac{k}{\delta} \right)^{d-1} \right\}$$

for a constant  $C \in \mathbb{R}^+$  that depends only on the sub-Gaussian norm of the i.i.d.  $\Omega_i$ -entries.

The proof of Theorem 3.5 largely follows the argument proposed in Section 2 of [1] concerning the so-called  $p$ -moment JL property of Khatri-Rao structured measurements. We note that Kronecker products of sub-Gaussian vectors are not sub-Gaussian in general, so the general idea is to use Markov's inequality for higher moments of the norm of  $\Omega_{-j}\mathbf{y}$  with a fixed  $\mathbf{y} \in \mathbb{R}^n$  to obtain the desired result. To proceed with the argument, we will need the following two concentration results. Before stating the concentration results, we recall a standard definition that the  $p$ -th root of the (absolute) moment of a random variable is referred to as the  $L^p$ -norm of the random variable, and for random variable  $X$  is denoted

$$\|X\|_{L^p} := (\mathbb{E}|X|^p)^{1/p} \quad (3.15)$$

this is of course because the expectation of a random variable in a probability space is by definition the Lebesgue integral of the function from the probability space to the reals, and so all the theorems about Lebesgue integration and metric spaces apply to random variables. Now onto our needed concentration results:

**Lemma 3.5.** *[Lemma 19 in [33]] Let  $\mathcal{Y}$  be a  $d - 1$  mode tensors with side lengths of size  $n$ ,  $p \geq 1$ , and  $\Omega_j(i, :) \in \mathbb{R}^n$  for  $j \in [d - 1], i \in [m]$  be independent random vectors each satisfying the Khintchine inequality  $\|\langle \Omega_j(i, :), \mathbf{y} \rangle\|_{L^p} \leq C_p \|\mathbf{y}\|_2$  for any vector  $\mathbf{y} \in \mathbb{R}^n$  where  $C_p$  a constant depending only on  $p$ . Then*

$$\|\langle \Omega_1(i, :) \otimes \Omega_2(i, :) \otimes \dots \otimes \Omega_{d-1}(i, :), \text{vec}(\mathcal{Y}) \rangle\|_{L^p} \leq C_p^{d-1} \|\mathcal{Y}\|_2.$$

**Lemma 3.6.** [Corollary 2 in [41]] If  $p \geq 2$  and  $Z, Z_1, \dots, Z_m$  are i.i.d symmetric random variables then we have

$$\left\| \sum_{i=1}^m Z_i \right\|_{L^p} \leq C \sup_{s \in [\max\{2, \frac{p}{m}\}, p]} \left\{ \frac{p}{s} \left( \frac{m}{p} \right)^{1/s} \|Z\|_{L^s} \right\}.$$

Here  $C > 0$  is an absolute constant.

In particular, we will utilize the following corollary of Lemma 3.6.

**Corollary 3.3.** If under conditions of Lemma 3.6, in addition, we know that  $\|Z\|_{L^s} \leq (Cs)^{d-1}$ , then

$$\left\| \frac{1}{m} \sum_{i=1}^m Z_i \right\|_{L^p} \leq C^{d-1} \max \left\{ 2^{d-1} \sqrt{\frac{p}{m}}, \frac{(pe)^{d-1}}{m} \right\}.$$

*Proof.* The proof of this corollary loosely follows the argument presented in [1]. Since  $\|Z\|_{L^s} \leq (Cs)^{d-1}$ ,  $Z_i \sim Z$ , the expression over which we are taking the supremum in Lemma 3.6 is a function whose derivative with respect to  $s$  is non-decreasing. That is, the derivative

$$\frac{\partial}{\partial s} \left[ \frac{p}{s} \left( \frac{m}{p} \right)^{1/s} s^{d-1} \right] = ps^{d-4} \left( \frac{m}{p} \right)^{1/2} \left( (d-2)s - \log \frac{m}{p} \right)$$

has at most a single root in the interval of interest at  $s = \frac{\log \frac{m}{p}}{d-2}$ . Noting the sign change at this root, we conclude that the maximum value must occur at the endpoints of the interval, and cannot occur at the critical point that is interior to the interval. Evaluating the function of interest at  $s = 2$  we obtain  $2^{d-2} \sqrt{mp}$ ; we will further upper-bound this by  $2^{d-1} \sqrt{mp}$  in order to simplify analysis for the right endpoint. Additionally, since we are interested only in an upper bound, we need not evaluate the possible endpoint  $s = \frac{p}{m}$ , since if  $2 < \frac{p}{m}$ , we are increasing the interval over which we are maximizing by instead considering  $s = 2$ .

We will now bound the expression at the right endpoint when  $s = p \geq 2$ . Clearly  $(1/p)^{1/p} \leq 1$  thus

$$\left( \frac{m}{p} \right)^{1/p} p^{d-1} \leq m^{1/p} p^{d-1}.$$

If the function value at the right endpoint actually dominates  $2^{d-1} \sqrt{mp}$  (i.e., our upper bound of the function value at the left endpoint), we will have that  $m^{1/p} p^{d-1} \geq 2^{d-1} \sqrt{mp}$  must hold. We will now use this assumption to remove the dependence on  $m$  in our current upper bound for the

function value at the right endpoint after noting that doing so will still yield a valid upper bound whenever the functions value at the right endpoint fails to already be bounded by  $2^{d-1}\sqrt{mp}$ .

Proceeding as planned, our assumption yields that

$$m^{1/2-1/p} \leq \left(\frac{p}{2}\right)^{d-1} p^{-1/2}.$$

Rearranging of terms, we get that

$$\begin{aligned} m^{1/p} &\leq \left[ \left(\frac{p}{2}\right)^{d-1} p^{-1/2} \right]^{\frac{2}{p-2}} \\ &= \left[ \left(\frac{p}{2}\right)^{\frac{2d-2}{p-2}} \right] p^{\frac{-1}{p-2}}. \end{aligned}$$

The factor  $p^{\frac{-1}{p-2}}$  is less than one. A tedious calculation reveals that  $\left(\frac{p}{2}\right)^{\frac{2d-2}{p-2}}$  is decreasing for all  $p \geq 2$ , and therefore  $m^{1/p}$  bounded by  $\lim_{p \rightarrow 2} \left(\frac{p}{2}\right)^{\frac{2d-2}{p-2}} = e^{d-1}$ . Maximizing over our two upper bounds and then averaging over  $m$ , we obtain the desired inequality.  $\square$

Now we are ready to give a formal proof of Theorem 3.5.

*Proof of Theorem 3.5:* Note that the result is unchanged for any choice of mode to leave out, so we will shorten the notation and work with  $\Omega := \Omega_{-j}$  within this proof. Let  $K$  be the sub-Gaussian norm of an entry in the  $\Omega_j$ 's, as in Definition 1.22. We aim to bound the probability that

$$\left| \frac{1}{m} \|\Omega \mathbf{x}\|_2^2 - \|\mathbf{x}\|_2^2 \right| \geq \epsilon \|\mathbf{x}\|_2^2 \quad \text{for a fixed } \mathbf{x} \in \mathbb{R}^{n^{d-1}}.$$

Without loss of generality, assume  $\|\mathbf{x}\|_2 = 1$ . Furthermore, note that

$$\frac{1}{m} \|\Omega \mathbf{x}\|_2^2 - 1 = \frac{1}{m} \sum_{i=1}^m [\langle \Omega(i, \cdot), \mathbf{x} \rangle^2 - 1]. \quad (3.16)$$

Now, since the entries of each  $\Omega_j$  are i.i.d. mean zero and variance one sub-Gaussian random variables, they satisfy Khintchine's inequality (see, e.g., [63]) in the form

$$\|\langle \Omega_j(i, \cdot), \mathbf{y} \rangle\|_{L^p} \leq CK\sqrt{p} \|\mathbf{y}\|_2 \quad \text{for any fixed } \mathbf{y} \in \mathbb{R}^n.$$

By Lemma 3.5, this implies that the rows of  $\Omega$  satisfy a generalized Khintchine's inequality in the form

$$\|\langle \Omega(i, \cdot), \mathbf{x} \rangle\|_{L^p} = \|\langle \Omega_1(i, \cdot) \otimes \Omega_2(i, \cdot) \otimes \dots \otimes \Omega_{d-1}(i, \cdot), \mathbf{x} \rangle\|_{L^p} \leq (C'p)^{\frac{d-1}{2}} \|\mathbf{x}\|_2, \quad (3.17)$$

where  $C'$  is a new constant that only depends on  $K$ .

To bound the  $L^p$ -norm of the sum in (3.16) we will now bound the  $L^p$ -norm of each summand. Using the centering Lemma 3.13 and continuing to estimate for one term we see that

$$\begin{aligned} \|\langle \Omega(i, \cdot), \mathbf{x} \rangle^2 - 1\|_{L^p} &\leq 2 \|\langle \Omega(i, \cdot), \mathbf{x} \rangle^2\|_{L^p} = 2 \|\langle \Omega(i, \cdot), \mathbf{x} \rangle\|_{L^{2p}}^2 \\ &= 2 \|\langle \Omega_1(i, \cdot) \otimes \Omega_2(i, \cdot) \otimes \dots \otimes \Omega_{d-1}(i, \cdot), \mathbf{x} \rangle\|_{L^{2p}}^2 \\ &\stackrel{(3.17)}{\leq} 2((C'2p)^{\frac{d-1}{2}})^2 \|\mathbf{x}\|_2^2 \leq (C''p)^{d-1} \|\mathbf{x}\|_2^2 = (C''p)^{d-1}. \end{aligned}$$

We would now like to apply Corollary 3.3 to help bound the  $L^p$ -norm of the sum in (3.16). However, we need to symmetrize our random variables first. Toward that end, define

$$Z_i = \rho_i \left( \langle \Omega(i, \cdot), \mathbf{x} \rangle^2 - 1 \right), \quad (3.18)$$

where  $\rho_i$  are i.i.d. Rademacher random variables. Note that  $\|Z_i\|_{L^p} = \|(\langle \Omega(i, \cdot), \mathbf{x} \rangle^2 - 1)\|_{L^p} \leq (C''p)^{d-1}$ .

Appealing now to Corollary 3.3 we have that,

$$\left\| \frac{1}{m} \sum_{i=1}^m Z_i \right\|_{L^p} \leq (C'')^{d-1} \max \left\{ 2^{d-1} \sqrt{\frac{p}{m}}, \frac{(pe)^{d-1}}{m} \right\} \quad (3.19)$$

for any  $p \geq 2$ . The upper bound in [43, Lemma 6.3] then further implies that

$$\left\| \frac{1}{m} \sum_{i=1}^m [\langle \Omega(i, \cdot), \mathbf{x} \rangle^2 - 1] \right\|_{L^p} \leq 2 \left\| \frac{1}{m} \sum_{i=1}^m Z_i \right\|_{L^p} \leq 2(C'')^{d-1} \max \left\{ 2^{d-1} \sqrt{\frac{p}{m}}, \frac{(pe)^{d-1}}{m} \right\}.$$

Employing Markov's inequality, we finally have that

$$\begin{aligned} \mathbb{P} \left\{ \left| \frac{1}{m} \sum_{i=1}^m [\langle \Omega(i, \cdot), \mathbf{x} \rangle^2 - 1] \right| \geq \epsilon \right\} &= \mathbb{P} \left\{ \left| \frac{1}{m} \sum_{i=1}^m [\langle \Omega(i, \cdot), \mathbf{x} \rangle^2 - 1] \right|^p \geq \epsilon^p \right\} \\ &\leq (C''')^{p(d-1)} \frac{\max \left\{ \left(\frac{p}{m}\right)^{p/2}, \frac{(pe)^{p(d-1)}}{m^p} \right\}}{\epsilon^p}. \end{aligned}$$

Taking  $p = \log(k/\delta)$  and  $m \geq \tilde{C}^{d-1} \max \left\{ \epsilon^{-2} \log \frac{k}{\delta}, \epsilon^{-1} \left( \log \frac{k}{\delta} \right)^{d-1} \right\}$ , the last expression is upper bounded by  $\delta/k$ . Hence,  $\Omega$  is an  $(\epsilon, \delta, k)$ -JL by the union bound over  $k$  vectors.  $\square$

*Remark 3.4.* In order to employ Lemmas 3.5 and 3.6, it is necessary that the rows  $\Omega_j(i, :)$  have independent and identical distributions and that these rows satisfy Khintchine’s inequality. Assuming that the matrices  $\Omega_i$  all have i.i.d sub-Gaussian *entries* as we have done in Theorem 3.5 implies both these necessary properties of the rows. However, we note that more general (though perhaps less natural) assumptions will also suffice. For example, the distributions of the i.i.d sub-Gaussian entries of the  $\Omega_i$  may also vary by column.

We can now use Theorem 3.5 to derive row bounds that guarantee that our Khatri-Rao structured sub-Gaussian measurement matrices will provide PCP sketches with high probability.

**Theorem 3.6.** *Let  $\epsilon > 0, 0 < \delta \leq e^{-2}$ , and  $X$  be a  $d$  mode tensor with side-lengths equal to  $n$ . Furthermore, suppose that  $\Omega_{-j} := \frac{1}{m} \Omega_1 \cdot \Omega_2 \cdot \dots \cdot \Omega_{j-1} \cdot \Omega_{j+1} \cdot \dots \cdot \Omega_d$  where the  $\Omega_i \in \mathbb{R}^{m \times n}$  for  $i \in [d] \setminus \{j\}$  are as in Theorem 3.5 with*

$$m \geq C^{d-1} \max \left\{ \epsilon^{-2} \log \frac{\left(\frac{141}{\epsilon}\right)^r}{\delta/2}, \epsilon^{-1} \left( \log \frac{\left(\frac{141}{\epsilon}\right)^r}{\delta/2} \right)^{d-1}, \frac{r}{\epsilon^2} \log \frac{17n^2}{\delta/2}, \frac{r}{\epsilon} \left( \log \frac{17n^2}{\delta/2} \right)^{d-1} \right\} \quad (3.20)$$

for a positive constant  $C \in \mathbb{R}^+$ . Then,  $\tilde{X}_{[j]} = X_{[j]} \Omega_{-j}^T$  will be an  $(\epsilon, 0, r)$ -PCP sketch of  $X_{[j]}$  with probability at least  $1 - \delta$ .

*Proof.* By Lemma 3.1 we know that it suffices for the measurement matrix  $\Omega_{-j}$  to have both the  $\left(\frac{\epsilon}{6}, \frac{\delta}{2}, \left(\frac{141}{\epsilon}\right)^r\right)$ -JL property and  $\left(\frac{\epsilon}{6\sqrt{r}}, \frac{\delta}{2}, 16n^2 + n\right)$ -JL property. Combining these two required JL properties with Theorem 3.5 yields (3.20) after combining and simplifying constants.  $\square$

We can now see that both Kronecker and Khatri-Rao structured measurements can satisfy the PCP property required by Theorem 3.3. This demonstrates that such memory-efficient measurements may be used to bound the first term in (3.9) as desired. Given this partial success, we will now turn our attention to the second term in (3.9).

### 3.2.4 Bounding $\|\mathcal{X}_1 - \mathcal{X}_2\|_2$

Next, we show how to bound Term II in (3.9). Recall that  $\mathcal{X}_1$  is the output of Algorithm 3.1, and that  $\mathcal{X}_2$  is the tensor recovered by the two-pass algorithm consisting of the first “Factor matrix recovery” phase of Algorithm 3.1 followed by the second pass core recovery procedure discussed in Section 3.2.1. That is,  $\mathcal{X}_1 := [[\mathcal{H}, Q_1, \dots, Q_d]]$  is the single-pass estimate of the tensor  $\mathcal{X}$  output

by Algorithm 3.1, and

$$\mathcal{X}_2 = \mathcal{G} \times_1 Q_1 \times_2 \cdots \times_d Q_d = \mathcal{X} \times_1 Q_1 Q_1^T \times_2 \cdots \times_d Q_d Q_d^T,$$

where  $\mathcal{G}$  is the core estimate from the two-pass algorithm, and where the  $Q_i \in \mathbb{R}^{n \times r}$  have  $r$  orthonormal columns.

To begin, we note that the one-pass core  $\mathcal{H}$  computed by Algorithm 3.1 can be recovered from its input measurements by

$$\begin{aligned} \mathcal{H} &= \mathcal{B}_c \times_1 (\Phi_1 Q_1)^\dagger \times_2 \cdots \times_d (\Phi_d Q_d)^\dagger \\ &= (\mathcal{X} \times_1 \Phi_1 \times_2 \cdots \times_d \Phi_d) \times_1 (\Phi_1 Q_1)^\dagger \times_2 \cdots \times_d (\Phi_d Q_d)^\dagger. \end{aligned} \quad (3.21)$$

In addition, we note that the norm of the difference between the two estimates is the same as the norm of the difference of their cores since factor matrices have orthonormal columns.

**Lemma 3.7.** *In the notation outlined above,*

$$\|\mathcal{X}_1 - \mathcal{X}_2\|_2 = \|\mathcal{H} - \mathcal{G}\|_2.$$

*Proof.*

$$\begin{aligned} \|\mathcal{X}_1 - \mathcal{X}_2\|_2 &= \|\mathcal{H} \times_1 Q_1 \times_2 \cdots \times_d Q_d - \mathcal{G} \times_1 Q_1 \times_2 \cdots \times_d Q_d\|_2 \\ &= \|(\mathcal{H} - \mathcal{G}) \times_1 Q_1 \times_2 \cdots \times_d Q_d\|_2 \\ &= \|(Q_1 \otimes \cdots \otimes Q_d) \text{vec}(\mathcal{H} - \mathcal{G})\|_2 \\ &= \|\mathcal{H} - \mathcal{G}\|_2 \end{aligned}$$

since  $(Q_1 \otimes \cdots \otimes Q_d)$  has orthonormal columns. □

In order to simplify the presentation of our culminating results we next state a definition for an Affine-Embedding property. In Lemma 3.8 below we then describe how this property relates to the OSE and AMM properties.

**Definition 3.4** ( $(\epsilon, r, N)$ -**AE property**). Let  $\epsilon > 0$ , and  $r \in \mathbb{N}$ , Fix  $Q \in \mathbb{R}^{n \times r}$  an arbitrary matrix with orthonormal columns and an arbitrary matrix  $B \in \mathbb{R}^{n \times N}$ . A matrix  $\Phi \in \mathbb{R}^{m \times n}$  is an  $(\epsilon, r, N)$ -Affine Embedding (AE) for given matrices  $Q$  and  $B$  if it satisfies

$$\|(\Phi Q)^\dagger \Phi B\|_F \leq (1 + \epsilon) \|B\|_F. \quad (3.22)$$

With this definition in hand, we are now able to prove the main theorem of this section. It will allow us to relate Term II of (3.9) with Term I.

**Theorem 3.7.** Let  $\mathcal{X}_2 = [[\mathcal{G}, Q_1, \dots, Q_d]]$  denote the two-pass tensor estimate, and  $\mathcal{X}_1 = [[\mathcal{H}, Q_1, \dots, Q_d]]$  denote the single-pass tensor estimate for a  $d$ -mode tensor  $\mathcal{X}$ . Furthermore, let  $\epsilon \in (0, 1)$ , and  $\Phi_i \in \mathbb{R}^{m_c \times n}$  be a  $(\epsilon/d, r, n^{i-1}r^{d-i})$ -AE for the matrices  $Q_i$  and  $(X_{[i]} - (X_2)_{[i]}) \otimes_{j=1}^{i-1} I_n \otimes_{j=i+1}^d (\Phi_j Q_j)^\dagger \Phi_j$  for each  $i \in [d]$ . Then,

$$\|\mathcal{X}_1 - \mathcal{X}_2\|_2 \leq e^\epsilon \|\mathcal{X} - \mathcal{X}_2\|_2. \quad (3.23)$$

*Proof.* By Lemma 3.7 it is enough to estimate the difference between the cores  $\mathcal{G}$  and  $\mathcal{H}$ . He have that

$$\begin{aligned} \mathcal{H} - \mathcal{G} &\stackrel{(3.21)}{=} (\mathcal{X} \times_1 \Phi_1 \times_2 \cdots \times_d \Phi_d) \times_1 (\Phi_1 Q_1)^\dagger \times_2 \cdots \times_d (\Phi_d Q_d)^\dagger - \mathcal{G} \\ &= ((\mathcal{X} - \mathcal{X}_2) \times_1 \Phi_1 \times_2 \cdots \times_d \Phi_d) \times_1 (\Phi_1 Q_1)^\dagger \times_2 \cdots \times_d (\Phi_d Q_d)^\dagger \\ &\quad + (\mathcal{X}_2 \times_1 \Phi_1 \times_2 \cdots \times_d \Phi_d) \times_1 (\Phi_1 Q_1)^\dagger \times_2 \cdots \times_d (\Phi_d Q_d)^\dagger - \mathcal{G} \\ &= (\mathcal{X} - \mathcal{X}_2) \times_1 (\Phi_1 Q_1)^\dagger \Phi_1 \times_2 \cdots \times_d (\Phi_d Q_d)^\dagger \\ &\quad + \mathcal{G} \times_1 (\Phi_1 Q_1)^\dagger \Phi_1 Q_1 \cdots \times_d (\Phi_d Q_d)^\dagger \Phi_d Q_d - \mathcal{G} \\ &= (\mathcal{X} - \mathcal{X}_2) \times_1 (\Phi_1 Q_1)^\dagger \Phi_1 \times_2 \cdots \times_d (\Phi_d Q_d)^\dagger \Phi_d. \end{aligned}$$

Now consider the following related mode- $i$  unfolding for  $i \in [d]$ , where  $(\Phi_j Q_j)^\dagger \Phi_j$  for  $j < i$  is replaced with an  $n \times n$  identity matrix  $I_n$ ,

$$X'_i := (X_{[i]} - (X_2)_{[i]}) \otimes_{j=1}^{i-1} I_n \otimes_{j=i+1}^d (\Phi_j Q_j)^\dagger \Phi_j. \quad (3.24)$$

Each  $\Phi_i$  is an  $(\epsilon/d, r, n^{i-1}r^{d-i})$ -AE, where  $Q \leftarrow Q_i$  and  $B \leftarrow X'_i$ , for  $i = 1, 2, \dots, d$ . Thus,

$$\begin{aligned} \|\mathcal{H} - \mathcal{G}\|_2 &= \left\| (\Phi_1 Q_1)^\dagger \Phi_1 (X_{[1]} - (X_2)_{[1]}) \bigotimes_{j=2}^d (\Phi_j Q_j)^\dagger \Phi_j \right\|_F \\ &\leq \left(1 + \frac{\epsilon}{d}\right) \|X'_1\|_F \\ &\quad \vdots \\ &\leq \left(1 + \frac{\epsilon}{d}\right)^d \|X'_d\|_F \\ &\leq e^\epsilon \|\mathcal{X} - \mathcal{X}_2\|_2, \end{aligned}$$

where we have used Definition 3.4  $d$ -times together with the bound  $(1 + \frac{\epsilon}{d})^d \leq e^\epsilon$ .  $\square$

### 3.2.5 Putting it All Together with Row Bounds

In the previous subsections we have demonstrated that we can bound both error terms in (3.9) when the leave-one-out and core measurements satisfy certain embedding properties. In particular, we have shown how the Johnson-Lindenstrauss property (Definition 1.18) can be used to obtain both the Oblivious Subspace Embedding (Definition 1.23) and an Approximate Matrix Multiplication properties (Definition 1.25). These two properties are then used with compositions and direct sums to show how a tensor unfolding will satisfy a Projection Cost Preserving property (Definition 3.2), which was the essential ingredient in bounding the error term  $\|\mathcal{X} - \mathcal{X}_2\|_2$ . Next, we introduced Affine-Embeddings (Definition 3.4), which are the crucial ingredient needed for bounding the error term  $\|\mathcal{X}_1 - \mathcal{X}_2\|_2$ . In this section we will show how JL and OSE properties imply the AE property. All together then, these will enable us to verify the requirements of Theorem 3.8 in a straightforward manner once we have specified the type of leave-one-out measurements, and the particular type of sensing matrices.

**Theorem 3.8** (Error bound for one-pass). *Let  $\mathcal{X}_1 = [[\mathcal{H}, Q_1, \dots, Q_d]]$  denote the single-pass tensor estimate for a  $d$ -mode tensor  $\mathcal{X}$  with side length  $n$  obtained from Algorithm 3.1 using leave-one-out linear measurements  $B_i$  for  $i \in [d]$ , and core measurements  $\mathcal{B}_c$ . Let  $\epsilon > 0$  and  $\delta \in (0, 1/2)$ . Then,*

$$\|\mathcal{X}_1 - \mathcal{X}\|_2 \leq (1 + e^\epsilon) \sqrt{\frac{1 + \epsilon}{1 - \epsilon} \sum_{j=1}^d \Delta_{r,j}} \quad \text{where } \Delta_{r,j} := \sum_{i=r+1}^{\tilde{n}_j} \sigma_i(X_{[j]})^2 \quad (3.25)$$

will hold whenever

1.  $\Omega_{(i,i)} \in \mathbb{R}^{n \times n}$  are full-rank matrices for all  $i \in [d]$ ,
2.  $\Omega_{(i,i)}^{-1} B_i = X_{[i]} \Omega_{-j}^T \in \mathbb{R}^{n \times m^{d-1}}$  are  $(\epsilon, 0, r)$ -PCP sketches of  $X_{[i]}$  for each  $i \in [d]$ , and
3.  $\Phi_i \in \mathbb{R}^{m_c \times n}$  is an  $(\epsilon/d, r, n^{i-1} r^{d-i})$ -AE for the matrices  $Q_i$  and  $X'_i$  as in (3.24) for all  $i \in [d]$ .

*Proof.* Recalling (3.9) we have that

$$\|\mathcal{X}_1 - \mathcal{X}\|_2 = \|\mathcal{X}_1 - \mathcal{X} + \mathcal{X}_2 - \mathcal{X}_2\|_2 \leq \underbrace{\|\mathcal{X} - \mathcal{X}_2\|_2}_{\text{Term I}} + \underbrace{\|\mathcal{X}_1 - \mathcal{X}_2\|_2}_{\text{Term II}}. \quad (3.26)$$

We know from Theorem 3.3 that

$$\|\mathcal{X} - \mathcal{X}_2\|_2 \leq \sqrt{\frac{1 + \epsilon}{1 - \epsilon} \sum_{j=1}^d \Delta_{r,j}} \quad (3.27)$$

whenever  $X_{[j]} \Omega_{-j}^T \in \mathbb{R}^{n \times m^{d-1}}$  are  $(\epsilon, 0, r)$ -PCP sketches of  $X_{[j]}$  for each  $j \in [d]$ . Furthermore, Theorem 3.7 together with our third assumption implies that

$$\|\mathcal{X}_1 - \mathcal{X}_2\|_2 \leq e^\epsilon \|\mathcal{X} - \mathcal{X}_2\|_2. \quad (3.28)$$

Using (3.27) and (3.28) in (3.9) now yields the desired result.  $\square$

We now have need for the lemma that links the AE property to the JL and OSE properties, and thus provides the necessary machinery to fully account for row bounds that guarantee with high probability that the recovered tensor satisfies the stated bound in Theorem 3.8.

**Lemma 3.8.** *Let  $B \in \mathbb{R}^{n \times N}$  and suppose that  $Q \in \mathbb{R}^{n \times r}$ ,  $n \geq r$ , has orthonormal columns. If  $\Phi \in \mathbb{R}^{m \times n}$  is an  $(\frac{1}{2}, \delta, r)$ -OSE for  $Q$ , and has the  $(\frac{\epsilon}{2\sqrt{r}}, \delta)$ -AMM property for  $Q^T$  and  $(I - QQ^T)B$ , then  $\Phi$  will be an  $(\epsilon, r, N)$ -AE for the matrices  $Q$  and  $B$  with probability at least  $1 - 2\delta$ .*

*Proof.* Denote  $\tilde{Y} := (\Phi Q)^\dagger \Phi B$  and  $Y' := Q^T B$ , the solutions to the sketched and un-skeched linear least square problems given by minimizing  $\|\Phi B - \Phi Q Y\|_F$  and  $\|B - Q Y'\|_F$ , respectively, with respect to  $Y$ . Whenever  $\Phi$  is a  $(1/2, \delta, r)$ -OSE for  $Q$  we know that  $\Phi Q$  will be full-rank. It follows

that  $(\Phi Q)^T \Phi (B - Q\tilde{Y}) = 0$  will then also hold because  $(\Phi Q)^\dagger = [(\Phi Q)^T (\Phi Q)]^{-1} (\Phi Q)^T$ . As a consequence,

$$\begin{aligned} Q^T \Phi^T \Phi Q (\tilde{Y} - Y') &= Q^T \Phi^T \Phi Q (\tilde{Y} - Y') + (\Phi Q)^T \Phi (B - Q\tilde{Y}) \\ &= Q^T \Phi^T \Phi (Q\tilde{Y} - QY' + B - Q\tilde{Y}) \\ &= Q^T \Phi^T \Phi (B - QY'). \end{aligned}$$

When the approximate matrix multiplication property also holds we will now have that

$$\begin{aligned} \|(\Phi Q)^T \Phi Q (\tilde{Y} - Y')\|_F &= \|Q^T \Phi^T \Phi (B - QY')\|_F = \|Q^T \Phi^T \Phi (I - QQ^T) B\|_F \\ &\leq \frac{\epsilon}{2\sqrt{r}} \|Q^T\|_F \|(I - QQ^T) B\|_F \\ &= \frac{\epsilon}{2} \|(I - QQ^T) B\|_F = \frac{\epsilon}{2} \|B - QY'\|_F. \end{aligned} \tag{3.29}$$

Furthermore, whenever  $\Phi$  is a  $(\frac{1}{2}, \delta, r)$ -OSE for the column space of  $Q$ , all the eigenvalues of  $Q^T \Phi^T \Phi Q - I$  will be within the interval  $[-1/2, 1/2]$ . Thus, we can bound its operator norm  $\|Q^T \Phi^T \Phi Q - I\| \leq 1/2$ . We may now combine this operator norm bound with (3.29) and see that

$$\begin{aligned} \|\tilde{Y} - Y'\|_F &= \|(\tilde{Y} - Y') - (\Phi Q)^T \Phi Q (\tilde{Y} - Y') + (\Phi Q)^T \Phi Q (\tilde{Y} - Y')\|_F \\ &\leq \|(\Phi Q)^T \Phi Q (\tilde{Y} - Y')\|_F + \|[(\Phi Q)^T \Phi Q - I] (\tilde{Y} - Y')\|_F \\ &\leq \frac{\epsilon}{2} \|(I - QQ^T) B\|_F + \|[(\Phi Q)^T \Phi Q - I]\| \|\tilde{Y} - Y'\|_F \\ &\leq \frac{\epsilon}{2} \|(I - QQ^T) B\|_F + \frac{1}{2} \|\tilde{Y} - Y'\|_F. \end{aligned}$$

Rearranging the inequality above while noting the invariance of the Frobenius norm to multiplication by a matrix with orthogonal columns, we learn that

$$\|\tilde{Y} - Y'\|_F = \|Q(\tilde{Y} - Y')\|_F \leq \epsilon \|(I - QQ^T) B\|_F.$$

To finish, we may now apply the triangle inequality to see that

$$\begin{aligned}
\|\tilde{Y}\|_F &\leq \|\tilde{Y} - Y'\|_F + \|Y'\|_F \\
&\leq \epsilon \|(I - QQ^T)B\|_F + \|Q^T B\|_F \\
&\leq \epsilon \|(I - QQ^T)\| \|B\|_F + \|Q^T\| \|B\|_F \\
&= (1 + \epsilon) \|B\|_F.
\end{aligned}$$

In addition, we note that taking a union bound over the two necessary OSE and AMM conditions establishes the stated probability guarantee.  $\square$

We are now prepared to state how a particular choice of distribution used to generate our measurement matrices as well as the leave-one-out measurement type (Kronecker or Khatri-Rao) can satisfy the error bound (3.25) with high probability. Note that below Algorithms 3.9 and 3.11 refer to the specialization of Algorithm 3.1 to the type of leave-one-out measurement (Kronecker or Khatri-Rao, respectively).

**Theorem 3.9** (Error bound for one-pass Kronecker-structured sub-Gaussian measurements). *Suppose  $\mathcal{X}$  is a  $d$ -mode tensor with side length  $n$ . Let  $\epsilon > 0$ ,  $\delta \in (0, \frac{1}{3})$ ,  $r \in [n]$ . Furthermore, let*

1.  $\Omega_{(i,i)} \in \mathbb{R}^{n \times n}$  be arbitrary full-rank matrices,
2.  $\Omega_{(i,j)} \in \mathbb{R}^{m \times n}$  for  $i \neq j$  be random matrices with mutually independent, mean zero, variance  $m^{-1}$ , sub-Gaussian entries with

$$m \geq \max \left\{ \frac{C_1 r (d-1)^2}{\epsilon^2} \ln \left( \frac{n^d d^2}{\delta} \right), \frac{C_2 (d-1)^2}{\epsilon^2} \ln \left( \left( \frac{141}{\epsilon} \right)^r \frac{n^{d-2} d^2}{\delta} \right) \right\}, \text{ and}$$

3.  $\Phi_i \in \mathbb{R}^{m_c \times n}$  be random matrices with mutually independent, mean zero, variance  $m^{-1}$ , sub-Gaussian entries with

$$m_c \geq \max \left\{ C_3 \ln \left( \frac{(94)^r d}{\delta} \right), \frac{C_4 r d^2}{\epsilon^2} \ln \left( \frac{d(r + n^{d-1})^2}{\delta} \right) \right\}. \quad (3.30)$$

Then  $\mathcal{X}_1 = [[\mathcal{H}, Q_1, \dots, Q_d]]$  the output of Algorithm 3.9 (i.e., Algorithm 3.1 specialized to Kronecker sub-Gaussian measurements  $\mathcal{B}_i$  and  $\mathcal{B}_c$ ), will satisfy (3.25) with probability at least  $1 - 3\delta$ .

*Proof.* We verify that the requirements of Theorem 3.8 are satisfied. Note that:

1.  $\Omega_{(i,i)} \in \mathbb{R}^{n \times n}$  are full-rank matrices by assumption.
2. We have from Corollary 3.2 where  $\delta \leftarrow \frac{\delta}{d}$  that  $X_{[j]}\Omega_{-j}^T$  is a  $(\epsilon, 0, r)$ -PCP sketch of  $X_{[j]}$  for all  $j \in [d]$  with probability at least  $1 - \delta$  when  $\Omega_{(i,j)} \in \mathbb{R}^{m \times n}$  are independent sub-Gaussian random matrices with

$$m \geq \max \left\{ \frac{C_1 r (d-1)^2}{\epsilon^2} \ln \left( \frac{n^d d^2}{\delta} \right), \frac{C_2 (d-1)^2}{\epsilon^2} \ln \left( \left( \frac{141}{\epsilon} \right)^r \frac{n^{d-2} d^2}{\delta} \right) \right\}.$$

3. A substitution of  $\epsilon \leftarrow \frac{1}{2}$ ,  $\delta \leftarrow \frac{\delta}{d}$  into Corollary 1.1 yields,

$$m_c \geq C_3 \ln \left( \frac{(94)^r d}{\delta} \right) \quad (3.31)$$

in order to ensure  $\Phi_i$  is  $(\frac{1}{2}, \frac{\delta}{d}, r)$ -OSE. Using Corollary 1.2, where  $\epsilon \leftarrow \frac{\epsilon}{2d\sqrt{r}}$ ,  $\delta \leftarrow \frac{\delta}{d}$  and noting that the matrices  $Q_i$  and  $X'_i$  as in (3.24) have are  $r \times n$  and  $n \times n^{i-1} r^{d-i}$ , respectively, we have that when

$$m_c \geq \frac{C_4 r d^2}{\epsilon^2} \ln \left( \frac{2d(r + n^{d-1})^2}{\delta} \right) \quad (3.32)$$

then  $\Phi_i$  has the  $(\frac{\epsilon}{2d\sqrt{r}}, \frac{\delta}{d})$ -AMM property for each  $i \in [d]$ . Lemma 3.8 now shows how the OSE and AMM properties ensure that  $\Phi_i$  has the desired AE property for each  $i \in [d]$  with probability at least  $1 - 2\delta/d$ . The union bound now implies that the third requirement of Theorem 3.8 will hold with probability at least  $1 - 2\delta$ .

Taking a maximum over (3.31) and (3.32) after simplifying and adjusting constants then yields (3.30). A final union bound over the failure probabilities for requirements of  $\Omega_{(i,j)}$  and  $\Phi_i$  now yields the result.  $\square$

The following runtime analysis demonstrates that instances of Algorithm 3.1 can indeed recover low-rank approximations of  $d$ -mode tensors of side length  $n$  in  $o(n^d)$ -time. As a result, one can see that Algorithm 3.1 is effectively a sub-linear time recovery algorithm for a large class of low Tucker-rank tensors.

**Theorem 3.10.** *Suppose that  $m < n < m^{d-1}$  and that  $m > Cm_c$  for some absolute constant  $C > 0$ . Then, Algorithm 3.9, when given the  $B_i$  and  $\mathcal{B}_c$  measurements as input, runs in  $O(dn^2 m^{d-1})$ -time*

and requires the storage of  $dnm^{d-1} + m_c^d$  measurement tensor entries, and at most  $d(d-1)mn + n^2 + dm_c n$  total measurement matrix entries.

*Proof.* Inside the factor matrix recovery loop of Algorithm 3.5, called by Algorithm 3.9, the two main sub-tasks are to solve the linear system  $\Omega_{(i,i)}F_i = B_i$  and to compute a truncated SVD,  $F_i = U\Sigma V^T$ . Solving the linear system can be accomplished using  $QR$ -factorization via Householder orthogonalization. Doing so requires  $\frac{4}{3}n^3$  floating point operations to compute the factorization of  $\Omega_{(i,i)}$ , and  $2n^2m^{d-1}$  operations to form  $Q^T B_i$  and  $n^2m^{d-1}$  operations to solve  $RF_i = Q^T B_i$  via back substitution. The complexity of computing the SVD is  $O(nm^{d-1} \min(n, m^{d-1}))$ . Therefore the factor recovery loop has overall complexity

$$O(dn^2m^{d-1})$$

if we assume that  $n < m^{d-1}$ .

Next we consider the core recovery loop. for the first iteration of the loop of Algorithm 3.8, we must form  $\Phi_1 Q_1$  at a cost of  $O(m_c nr)$ . Next we solve a linear system  $\Phi_1 Q_1 H = B_1$  at a cost of  $O(2m_c r^2 - \frac{2}{3}r^3 + 3m_c^d r)$ . The first iteration dominates the complexity, since subsequent solves use a smaller right hand side formed from solutions from the previous iterations. Furthermore, if we assume  $m_c = O(m)$ , we have a core recovery loop with complexity

$$O(dm^d r).$$

Thus overall the recovery algorithm has  $O(dn^2m^{d-1})$  complexity. In the situation where  $\Omega_{(i,i)} = I_n$  then the computation of the SVDs in the factor recovery step dominates the run-time of the algorithm. Clearly the size of the measurement tensors are  $nm^{d-1}$  per factor and  $m_c^d$  for the core, which yields the space complexity of the measurements.  $\square$

One of the advantages of the structure of the argument in Theorem 3.8 is that once it is known how to ensure a given random matrix will satisfy the JL property, we can (with the help of Lemma 3.8) account for how to assemble related measurement operators that satisfy the error bound (3.25) in a straightforward way. For example, using Theorem 3.1 in [38] along with bounds appearing in that work on the sketching dimension, we have that a sub-sampled and scrambled

Fourier matrix is a  $(\epsilon, \delta, p)$ -JL of vectors in  $\mathbb{R}^n$  provided that

$$m \geq \frac{C}{\epsilon^2} \log\left(\frac{p}{\delta}\right) \log^4 n.$$

Using such existing results one can easily update Theorem 3.9 to instead use sub-sampled and scrambled Fourier measurement matrices  $\Omega_{(i,j)}$  and  $\Phi_i$  instead of matrices with independent sub-Gaussian entries.

Furthermore, it need not be the case that the distribution is the same for each component map  $\Omega_{(i,j)}$  or  $\Phi_i$ . It is important only that each map satisfies the JL primitive for arbitrary sets. We are free to choose a measurement map for a particular mode to suit some other purpose. For example, in the case that the side lengths of the tensor are unequal, we may prefer to choose a map that admits a fast matrix-vector multiply in order to economize run-time for the modes which are long, and on smaller modes, choose maps which have better trade-offs for quality of approximation in terms of  $m$  (e.g., we may prefer dense sub-Gaussian random matrices for these modes).

On the other hand, a measurement ensemble which does not rely on the Kronecker product of components, like that in Theorem 3.5 does not admit this sort of reasoning. Mixing measurement maps of different kinds in that case has no clear advantage, and indeed, may even serve to undermine the advantages. Nonetheless,

Theorem 3.8 can still be used to provide Khatri-Rao structured leave-one-out measurement results. Verifying the requirements of Theorem 3.8 for Khatri-Rao measurements using Theorem 3.6 yields the following result.

**Theorem 3.11** (Error bound for one-pass Khatri-Rao structured sub-Gaussian measurements).

*Suppose  $\mathcal{X}$  is a  $d$ -mode tensor with side length  $n$ , Let  $\epsilon > 0$ ,  $\delta \in (0, \frac{1}{3})$ ,  $r \in [n]$ . Furthermore, let*

1.  $\Omega_{(i,i)} \in \mathbb{R}^{n \times n}$  be full-rank matrices of any kind,
2.  $\Omega_{(i,j)} \in \mathbb{R}^{\tilde{m} \times n}$  for  $i \neq j$  be random matrices with mutually independent, mean zero, variance

$\tilde{m}^{-1}$ , sub-Gaussian entries with

$$\tilde{m} \geq C^{d-1} \max \left\{ \epsilon^{-2} \ln \left( \frac{\left(\frac{141}{\epsilon}\right)^r 2d}{\delta} \right), \epsilon^{-1} \left( \ln \left( \frac{\left(\frac{141}{\epsilon}\right)^r 2d}{\delta} \right) \right)^{d-1}, \right. \\ \left. \frac{r}{\epsilon^2} \ln \left( \frac{34n^2 d}{\delta} \right), \frac{r}{\epsilon} \left( \ln \left( \frac{34n^2 d}{\delta} \right) \right)^{d-1} \right\},$$

and

3.  $\Phi_i \in \mathbb{R}^{m_c \times n}$  be random matrices with mutually independent, mean zero, variance  $m^{-1}$ , sub-Gaussian entries with

$$m_c \geq \max \left\{ C_3 \ln \left( \frac{(94)^r d}{\delta} \right), \frac{C_4 r d^2}{\epsilon^2} \ln \left( \frac{d(r + n^{d-1})^2}{\delta} \right) \right\}.$$

Then  $\mathcal{X}_1 = [[\mathcal{H}, Q_1, \dots, Q_d]]$  the output of Algorithm 3.11 (i.e., Algorithm 3.1 specialized to Khatri-Rao sub-Gaussian leave-one-out measurements  $\mathcal{B}_i$  and Kronecker measurements  $\mathcal{B}_c$ ) will satisfy (3.25) with probability at least  $1 - 3\delta$ .

*Proof.* The proof is again based on verifying the requirements of Theorem 3.8.

1. The first requirement is satisfied by assumption.
2. The row requirement for the  $\Omega_{(i,j)}$  follows from an application of Theorem 3.6 with  $\delta \leftarrow \delta/d$ .  
As a result of doing so, we learn that the second requirement will be satisfied with probability at least  $1 - \delta$  after a union bound.
3. The third requirement is satisfied identically to the argument in the proof of Theorem 3.9.

The proof now concludes identically to the proof of Theorem 3.9.  $\square$

Comparing the Theorem 3.10, we note, e.g., that Theorem 3.11 will require the storage of  $d\tilde{m}n + m_c^d$  measurement tensor entries (where the  $\tilde{m}$  here is from the second condition of Theorem 3.11). Comparing this to the  $dnm^{d-1} + m_c^d$  measurements from Theorem 3.10 (where  $m$  here is as in Theorem 3.9) one can see that there are parameter regimes where Khatri-Rao structured measurements will lead to a smaller overall measurement budget.

### 3.3 Experiments

In this section we present numerical results that support our theoretical contributions and address practical trade-offs involved in the different choices for measurement type. Unless otherwise

specified, the tensors in the experiments are random three-mode cubic tensors, with side length  $n = 300$  and rank  $r = 10$ . We use the following procedure (same as in [60]) to generate low-rank tensors; the core's entries are uniformly and independently drawn from  $[0, 1]$  and the factors are formed by first sampling a standard normal distribution for each entry and then normalized and made orthogonal using  $QR$ -factorization. The data points in the plots are the mean of 100 independent trials. The parameter  $m$  refers to the sketching dimension for maps  $\Omega_{(i,j)} \in \mathbb{R}^{m \times n}$  used in recovering factor  $Q_i$ . For the left-out mode we remove the need to solve the full  $n \times n$  linear system by setting  $\Omega_{(i,i)}$  to be the  $n \times n$  identity matrix. The parameter  $m_c$  refers to the sketching dimension for  $\Phi_i \in \mathbb{R}^{m_c \times n}$  which are used in recovering the core  $\mathcal{H}$ .

In experiments with noise, the additive Gaussian noise tensor  $\mathcal{N}$  is scaled according to the desired signal to noise ratio and added to the true, (low-rank) tensor  $\mathcal{X}_0$ . That is,  $\mathcal{X} = \mathcal{X}_0 + \mathcal{N}$  is the observed, noisy tensor.

Signal to noise ratio (SNR) is calculated as  $10 \log_{10} (\|\mathcal{X}\|_2 / \|\mathcal{X}_0 - \mathcal{X}\|_2)$ . Relative error is calculated as  $(\|\hat{\mathcal{X}} - \mathcal{X}\|_2) / \|\mathcal{X}_0\|_2$  where  $\hat{\mathcal{X}}$  is the full estimated tensor.

### 3.3.1 Recovering Low-Rank Tensors

In this first simple experiment, we fix the signal to noise ratio at 30 decibels (dB) and vary the sketching dimension  $m$  to show the dependence on the accuracy of our estimate on the number of measurements. For each  $m$  we set  $m_c = 2m$ . Rank truncation is fixed at  $r = 10$ , which matches the rank of the true, noiseless tensor  $\mathcal{X}_0$ , see Figure 3.2. For the plot (b) in the figure, we have the maximum principal angle among the three estimated factors and true factors  $(Q_i, U_i)$  in degrees, see [35]. Note, there is no straightforward way to plot the relative error which is due to the factor estimates vs. the core estimate, because the decomposition will in general not be unique. However, since principal angle is invariant to non-singular transformations, plot (b) provides empirical evidence that the factor estimates alone are improving with sketching dimension. We note that for these low-rank tensors with noise, we are able to fit at or below the level of noise (relative error of 0.001) easily - evidently finding good rank 10 approximations to the (full-rank) noisy tensor  $\mathcal{X}$ .

### Noisy Recovery of Low Rank Tensors, SNR = 30 dB

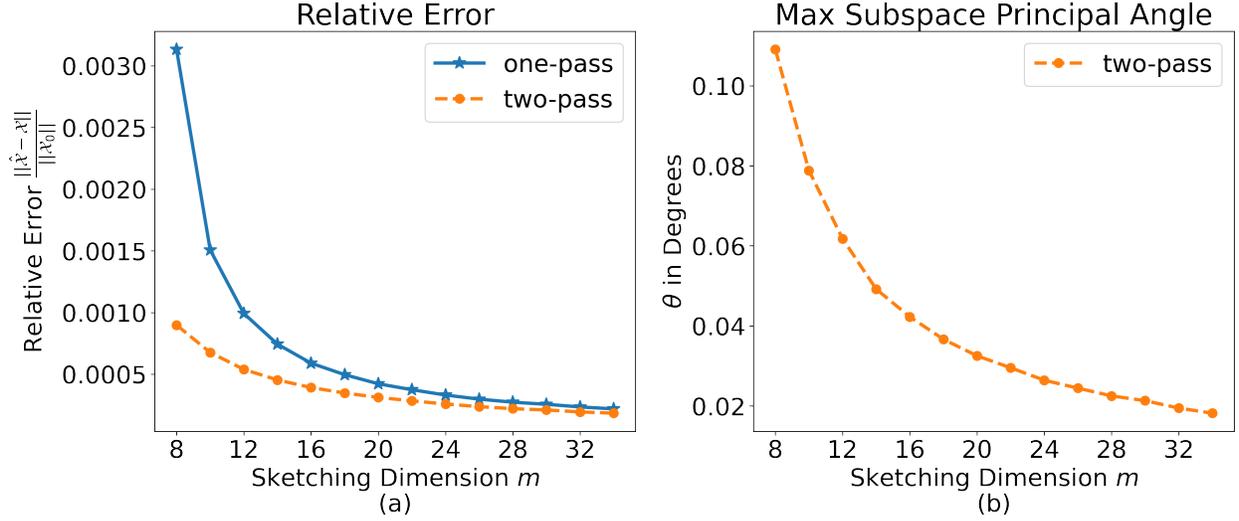


Figure 3.2 Error plots for different sketching dimensions and a fixed SNR of 30 dB and a fixed rank truncation of 10. Plot (a) compares relative errors for both one-pass and two-pass recovery. Plot (b) shows the maximum principal angle among all estimated sub-spaces and the true factor matrices.

This perhaps surprising result motivated us to try the method on a class of tensors in which we could be more certain about what quality of rank 10 approximation is achievable. In our second set of experiments, we examine performance on super-diagonal tensors with tail decay. Since we are truncating to rank 10, this tail can be thought of as structured, deterministic noise. These are tensors where all values are zero except for those on the diagonal, and where all values on the diagonal for indices larger than  $r = 10$ , we have some decay in their magnitude. In particular we have two types, exponential tail decay in plot (a), where

$$\mathcal{X}_{ijk} = \begin{cases} 1 & i = j = k \in [r + 1] \\ 10^{-1(i-r)} & i = j = k \in [r + 2, n] \\ 0 & \text{otherwise} \end{cases} \quad (3.33)$$

and polynomial tail decay in plot (b),

$$\mathcal{X}_{ijk} = \begin{cases} 1 & i = j = k \in [r + 1] \\ (i - r)^{-1} & i = j = k \in [r + 2, n] \\ 0 & \text{otherwise.} \end{cases} \quad (3.34)$$

These highly constrained tensors are clearly not low-rank, however it is reasonable to suppose that a recovery algorithm for a given rank truncation would output an estimate that is close to the leading  $r$  terms of the diagonal. The residual in that case will simply be the norm of the tail-sum  $\sqrt{\sum_{i=r+1}^n \chi_{iii}^2}$ , which we have included as the red horizontal line in Figure 3.3.

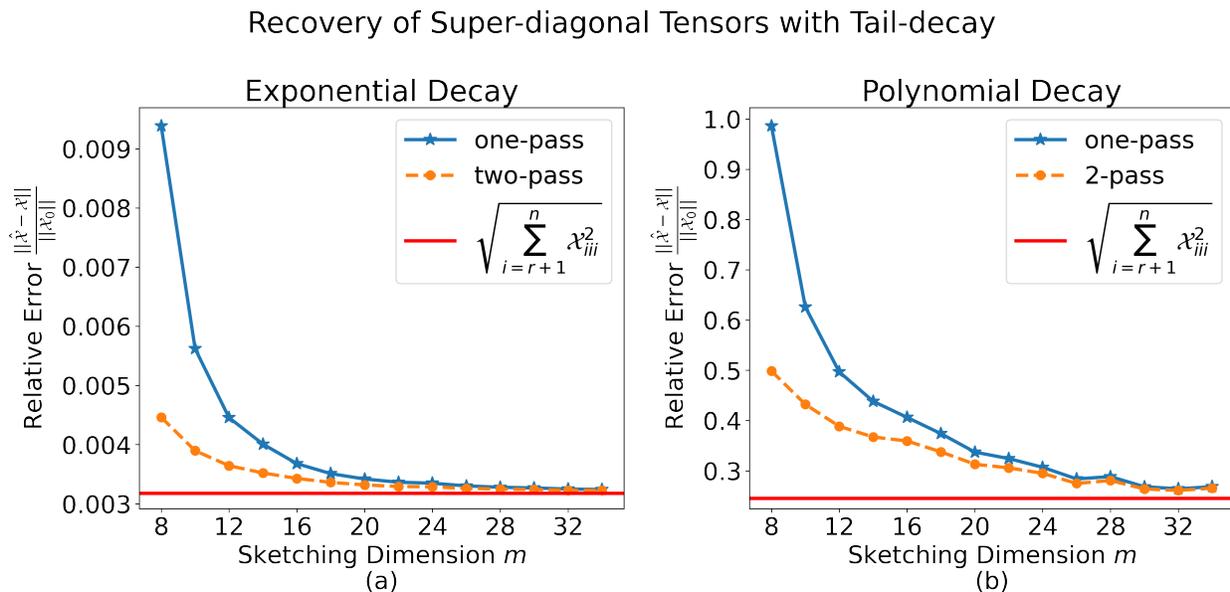


Figure 3.3 Error plots for different sketching dimensions in the noiseless setting with a fixed rank truncation of 10. Plot (a) compares relative errors for both one-pass and two-pass recovery for super-diagonal tensors where the diagonal entries have exponential decay of type (3.33), and plot (b) super-diagonal tensors with polynomial decay of the type (3.34).

### 3.3.2 Allocating Core and Factor Measurements

One question raised by our error analysis is how to weigh the error contribution between the tasks of estimating the factor matrices and estimating the core. In other words, for a given total measurement budget, how should we allocate between the two tasks if we wish to decrease overall relative error? In the following experiment (see Figure 3.4) we find the relative error under various noise levels for pairs of sketching dimensions  $(m, m_c)$ . We compare pairs  $(13, 12)$  and  $(11, 36)$  and  $(8, 48)$ . These choices of sketching dimensions were chosen since they have nearly equal overall compression ratios of 0.57%, 0.58%, 0.62% respectively, however they vary considerably on whether they emphasize measurements to be used in estimating the factors or the core of the tensor. Note that the two-pass error, which relies only on the factor matrix estimates is naturally

best when the factor sketches are larger, i.e. the  $m = 13$  case. However the relative error of the recovered tensor in the one-pass setting is more than ten times better when more of the total measurement budget is allocated to estimate the core as shown in Figure 3.4. This shows that in some situations it is preferable to allocate more resources to obtain measurements for the core than the factors, up to some threshold. For example in Figure 3.4, the rank of the true signal is 10, and going below this dimension for the factor sketches does correspond with no longer improving on the accuracy in terms of the trade-off between  $m$  and  $m_c$ .

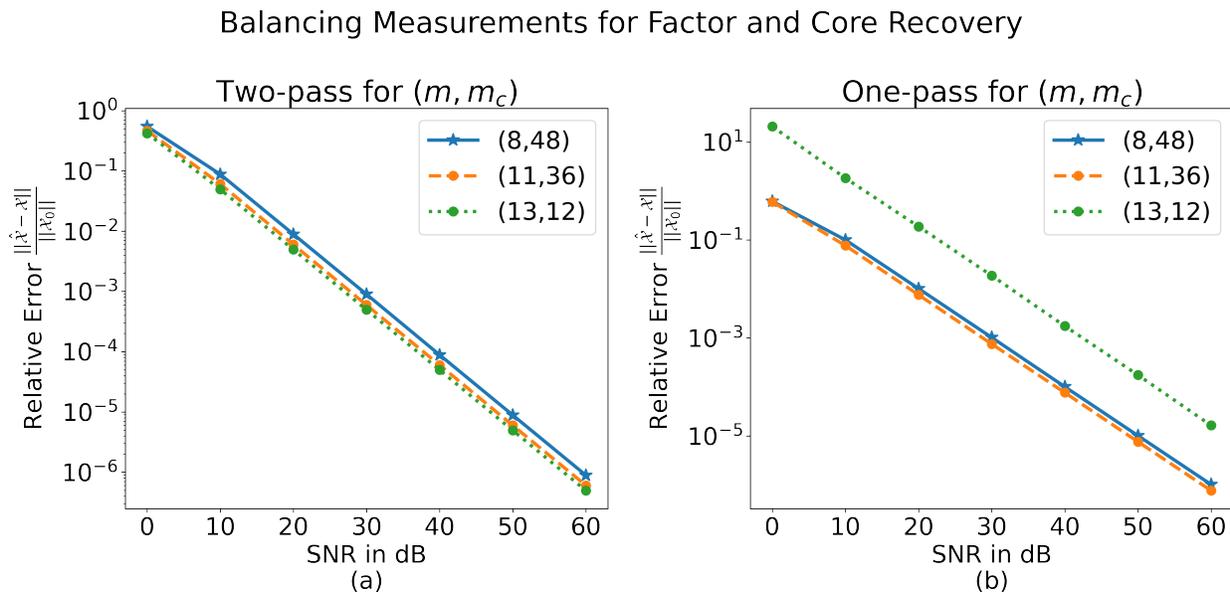


Figure 3.4 Relative error plots at different signal to noise ratios for two-pass and one-pass recovery of noisy low-rank tensors. Ordered pairs indicates the choice for sketching dimensions  $(m, m_c)$ .

### 3.3.3 Error Bounds Apply to sub-Gaussian Measurement Matrices

In this next experiment we demonstrate, in a similar manner as done in Figure 1 in [60], that recovery performance of Algorithm 3.11 does not vary greatly for different choices of types of sub-Gaussian measurement matrices. What is different from that earlier work is that the measurement ensembles are all Kronecker structured. Plotted in Figure 3.5 are relative errors for Gaussian (g), sparse uniform from  $[-1, 0, 1]$  with weights  $\frac{1}{6}, \frac{2}{3}, \frac{1}{3}$  (sp0), sub-sampled randomized Fourier transforms as in [64] (rfd) and a mixed measurement ensemble that uses Gaussian-RFD-sparse measurements where we vary by mode which measurement type is used, which is a scenario that is

practically and theoretically not well suited for the Khatri-Rao structured measurement operators used in [60].

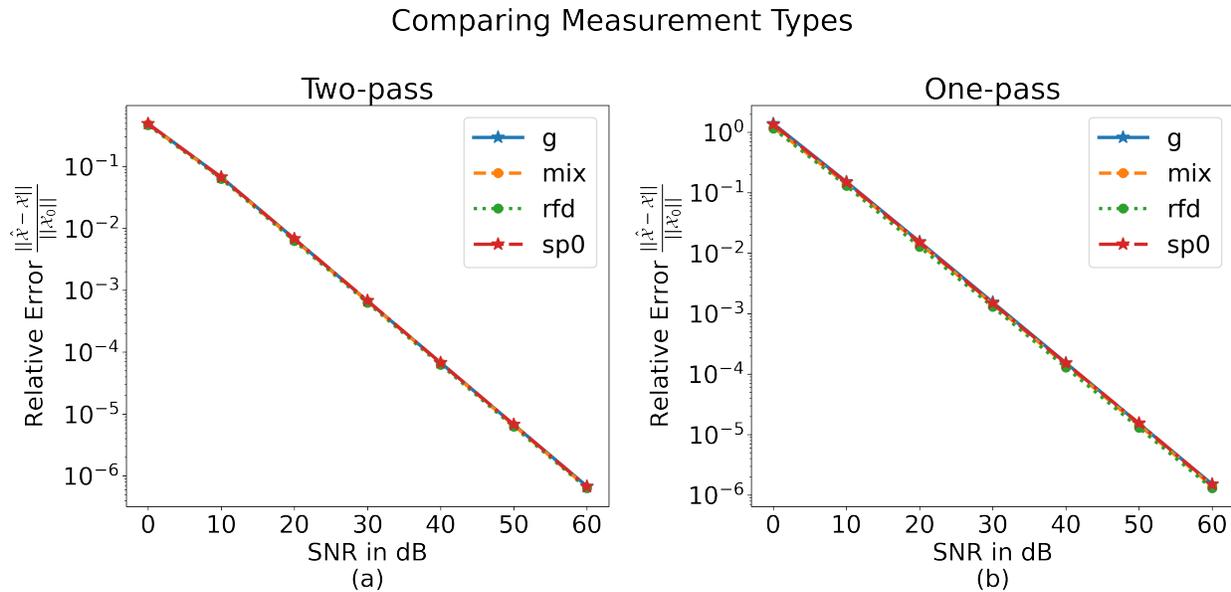


Figure 3.5 Relative errors for one-pass, two-pass for Kronecker measurement ensembles made up of different kinds of sub-Gaussian random matrices. The legend g, sp0, rfd, mix correspond to Gaussian, sparse, sub-sampled random Fourier transform, and a mixture of all three for the measurement ensembles.

### 3.3.4 Comparison to Khatri-Rao Structured Measurements

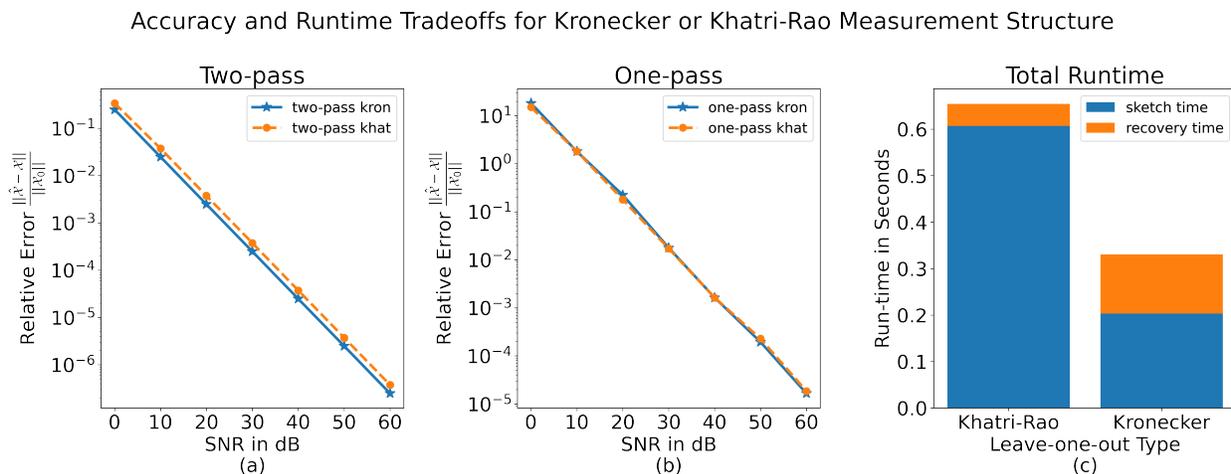


Figure 3.6 Relative error comparison for Khatri-Rao and Kronecker-structured measurements. Right hand figure shows the average time for sketching and recovery phases of the algorithm.

This set of experiments demonstrates that the sketching phase will dominate the run-time of

Algorithm 3.1 regardless of the choice of leave-one-out type, however the Kronecker-structured measurements are able to generate more measurements for a fixed number of operations as compared to Khatri-Rao structured measurements, see Figure 3.6. This means that it is possible to achieve similar or better performance using strictly modewise measurements and in less overall time as problems grow in size with respect to total number of tensor elements; i.e. both number of modes and length of those modes. In Figure 3.6 for the Kronecker-structured measurements we sketch to  $m = 25$ , for the Khatri-Rao ensemble we sketch pairs of modes to 225. We see that the Kronecker measurements perform incrementally better in terms of relative error but at less than half the overall run-time. Sketching times are about five times faster for the Kronecker-structured measurements as compared to the Khatri-Rao. Note that this does trade speed for space - the total number of entries in the leave one out measurements is nearly three times larger for the Kronecker-structured measurements versus the Khatri-Rao, i.e. sketches  $B_i$  as per (3.2) and (3.4) have sizes  $300 \times 25^2$ , and  $300 \times 225$  respectively.

### 3.3.5 Application to Video Summary Task

As a practical demonstration, we consider the same video summary task first described in [47] and again in [60]. In this demonstration, the video is taken with a camera in a fixed position. The video is a nature scene and a person walks in front of the camera at two different time points in the second half of the video. The first 100 and the last 193 frames are removed since they include setup that results in small shifts of the camera. The entire video has been converted to grayscale. This yields a three mode tensor of dimensions  $2200 \times 1080 \times 1980$  which has a size of about 41 GB when stored as an array of doubles. We wish to identify the parts of the scene that include the person walking and distinguish them from the relatively static scene elsewhere. As discussed in [60], there is a third salient time varying feature in this particular video, which is the light intensity of the scene, since at around frame 940 the scene darkens. Furthermore there are changes in the light intensity as the camera automatically adjusts after the person walks in and out of frame. For this reason, we cluster the frames using three centers, rather than two.

In all cases, we use  $k$ -means to cluster the frames, however we assign features to frames in four

different ways:

1. Using the sketch  $B_1 \in \mathbb{R}^{2200 \times 20^2}$ , as in (3.2) that leaves out the time dimension, then clustering using  $k$ -means on the rows of the unfolding of the sketch along the first, temporal mode.
2. Unfolding the temporal mode of the reconstructed tensor using a one-pass set of measurements, i.e.  $(X_1)_{[1]} \in \mathbb{R}^{2200 \times 2138400}$  (recall that  $X_1$  denotes the output of Algorithm 3.1).
3. Unfolding the reconstructed tensor using a two-pass scenario,  $(X_2)_{[1]} \in \mathbb{R}^{2200 \times 2138400}$  (recall that  $X_2$  denotes the output of Algorithm 3.10).
4. Estimated temporal factor matrix  $U_1 \in \mathbb{R}^{2200 \times 20}$  (see in Algorithm 3.1).

As we can see in Figures 3.7a, 3.7b, the sketch alone shows reliable clustering of the main temporal changes in the video, which verifies the observation in [60] about using the measurements as an effective feature set for clustering, although in that case the measurements were Khatri-Rao structured whereas ours are Kronecker-structured. The unfoldings of the reconstructed tensor also reliably distinguishes the main parts of the scene. The reconstruction is useful at least to get clustered interpretability. Although certainly natural to wish to cluster on the temporal factor, this method appears inferior to any of the preceding.

As an added advantage of using the modewise, Kronecker structured measurements, we can in principle select measurement maps for different modes. Gaussian measurement maps theoretically have some advantages over other types in terms of accuracy for a fixed number of measurements, whereas applying RFD or other Fourier-like transforms to modes that have longer fibers would net a better payoff in terms of overall run-time because of the faster matrix-vector multiply permitted by these structured matrices. In this demonstration, we use Gaussian matrices along the spatial modes, and *RFD* matrices for the temporal mode.

In the earlier work [47], the authors describe a variant of Tucker-Alternating Least Squares (aka Higher Order Orthogonal Iteration, multi-pass scenario) that employed TensorSketch to produce the necessary measurements used to reconstruct the same video tensor data we have used here. In the subsequent work [60], those authors again perform the same task, but use a single-pass approach which fits the framework we have described as Algorithm 3.1, where the measurement matrices

are Khatri-Rao structured, and the  $\Omega_i$  have entries drawn from standard Gaussian distribution. Furthermore, analysis of the type afforded by Theorem 3.11 may also explain the discrepancy between the sketching dimensions seen in [47] and [60]. Naturally there are several differences between the approaches, but the CountSketch matrices used in TensorSketch operators as shown in [33] have a  $\mathcal{O}\left(\frac{1}{\delta}\right)$  dependency in order to ensure the OSE property, whereas the other ensembles, such as dense Gaussians, enjoy an  $\mathcal{O}\left(\log \frac{1}{\delta}\right)$  dependence for this parameter.

As was discussed in [60], the video is not especially low-rank in practice - in particular along the spatial dimensions in terms of relative error of the reconstruction. However the clusters appear distinct enough that assigning clusters with this summary type of information is still possible.

### 3.4 Discussion

In this chapter we have described a measurement system that specializes the framework discussed in Section 1.5 and a simple recovery procedure that addresses the three main issues we noted about the two-stage measurement operators needed to ensure a TRIP, which was the key to make use of existing theory with regards to TIHT that was the subject of Chapter 2. That is, the leave-one-out alternative discussed in this chapter does not require working memory on the order of the size of the uncompressed signal, because it computes factors of the estimate directly from measurements. Furthermore, the recovery procedure is non-iterative, and provably recovers the tensor in the exact arithmetic, no noise, no rank truncation scenario, and has no reliance on unverifiable assumptions about the quality of the fit of a thresholding operator as TIHT had. On the third issue, whereby TRIP required an initial reshaping in order to achieve useful compression - the leave-one-out alternative trades this constraint for another; namely that one of the modes goes uncompressed. Direct comparison of which of these constraints is preferable is perhaps best left to a given application than any objective criteria that is data agnostic. It is worth mentioning however, that the need for the measurements to be independent, based on empirical evidence from synthetic data, is to a large extent motivated by the analysis. That is, practically speaking, one can reuse measurement maps to produce the different measurement tensors, or even share measurements that were used to estimate factors, and use them to estimate the core without any change in accuracy.

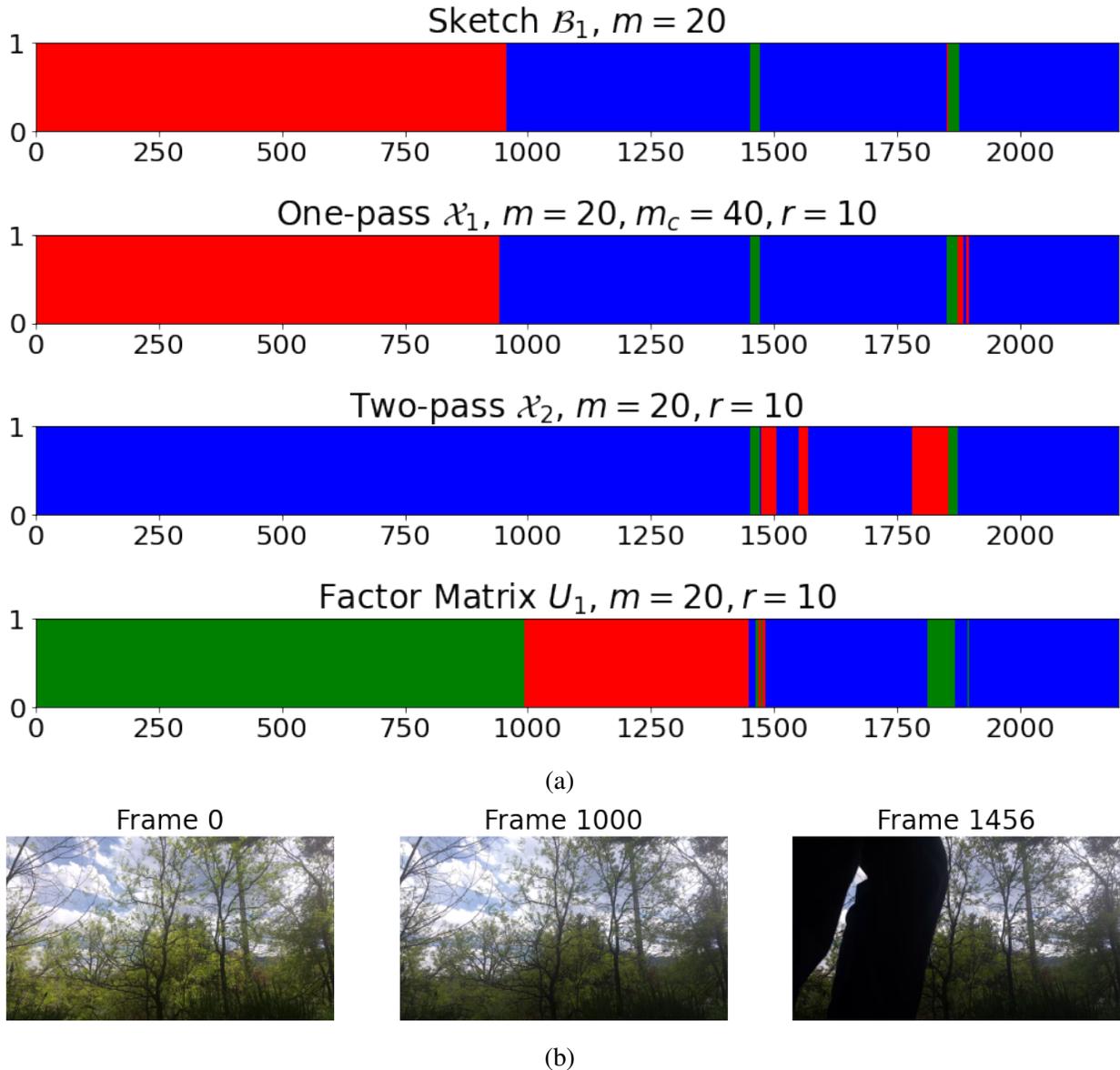


Figure 3.7 (a) Cluster assignments for the 2200 frames in the video. Top run corresponds to using the measurements for the first mode only  $B_1$ , middle rows use one and two pass approximations of the tensor, and the last row uses the factor matrix  $U_1$  for the temporal mode only. We use Gaussian sketching matrices for both spatial modes and the real part of  $RFD$  for the temporal mode. Sketching parameters are  $m = 20$ ,  $m_c = 40$  and rank truncation of  $r = 10$  in all modes. (b) Three reference frames at 0, 1000 and 1496.

The dependencies introduced however make the analysis of such a procedure considerably more complicated. A practitioner could economize on both the storage of the linear transformations, and number of measurements by reusing maps and “recycling” measurements.

Also, in this chapter we have shown how several variations can be unified under the leave-

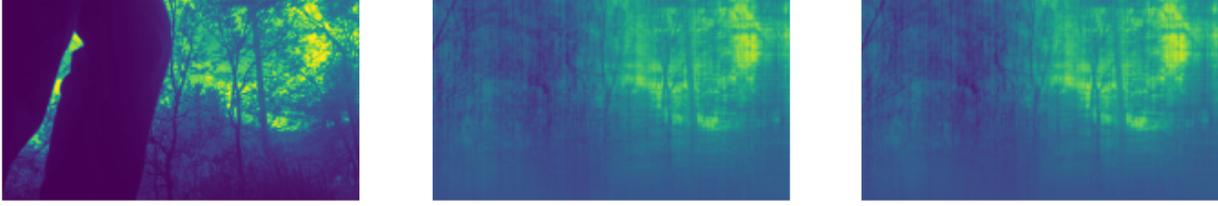


Figure 3.8 The 1456th frame of the grey-scale video is shown for the original, the one-pass, and two-pass reconstructions using sketching dimension of  $m = 300$ ,  $m_c = 601$  and  $r = 50$  for each mode. Although the reconstructions for this choice of sketching dimension and rank truncation are not particularly accurate for this video, nevertheless the reconstructions provide enough information to perform the summary task of clustering the frames into the major temporal changes that occur during the scene.

one-out concept, and the results made useful to analyze all of them. We have used empirical tests on simulated data to highlight important trade-offs in terms of what type of measurements were selected and how sensing matrices were structured. Finally, we demonstrated a simple, but representative application whereby clustering or segmenting video data is handled using the measurements. Recall the example from section 1.2.2 as to for why both the segmenting using the measurements, and the recovered tensor could be of practical value.

### 3.5 Technical Proofs

Herein we provide proofs for some technical lemmas needed to prove our main theorems, including a Lemma that we first stated in Chapter 1 regarding the AMM property. The second sub-section then addresses the proof of Theorem 3.2.

#### 3.5.1 Proof of Lemma 1.8

Our proof of Lemma 1.8 will utilize several intermediate lemmas. Our first lemma concerning the approximate preservation of inner products is a slight generalization of [2, Corollary 2].

**Lemma 3.9** (The JL property implies angle preservation). *Let  $S \subset \mathbb{C}^n$  with cardinality at most  $p$  and  $\epsilon \in (0, 1)$ . If a random matrix  $\Omega \in \mathbb{C}^{m \times n}$  has the  $(\epsilon/4, \delta, 4p^2)$ -JL property for*

$$S' = \left\{ \frac{\mathbf{x}}{\|\mathbf{x}\|_2} + \frac{\mathbf{y}}{\|\mathbf{y}\|_2}, \frac{\mathbf{x}}{\|\mathbf{x}\|_2} - \frac{\mathbf{y}}{\|\mathbf{y}\|_2}, \frac{\mathbf{x}}{\|\mathbf{x}\|_2} + i\frac{\mathbf{y}}{\|\mathbf{y}\|_2}, \frac{\mathbf{x}}{\|\mathbf{x}\|_2} - i\frac{\mathbf{y}}{\|\mathbf{y}\|_2} \mid \mathbf{x}, \mathbf{y} \in S \right\},$$

then

$$|\langle \Omega \mathbf{x}, \Omega \mathbf{y} \rangle - \langle \mathbf{x}, \mathbf{y} \rangle| \leq \epsilon \|\mathbf{x}\|_2 \|\mathbf{y}\|_2 \quad \forall \mathbf{x}, \mathbf{y} \in S \quad (3.35)$$

will be satisfied with probability at least  $1 - \delta$ .

*Proof.* Note that if either  $\mathbf{x} = \mathbf{0}$  or  $\mathbf{y} = \mathbf{0}$ , then (3.35) automatically holds because  $0 \leq 0$ . Thus, suppose without loss of generality that  $\mathbf{x}, \mathbf{y} \neq \mathbf{0}$ . Considering the normalizations  $\mathbf{u} = \frac{\mathbf{x}}{\|\mathbf{x}\|_2}, \mathbf{v} = \frac{\mathbf{y}}{\|\mathbf{y}\|_2}$ , one can see that the polarization identity implies that

$$\begin{aligned} |\langle \Omega \mathbf{u}, \Omega \mathbf{v} \rangle - \langle \mathbf{u}, \mathbf{v} \rangle| &= \left| \frac{1}{4} \sum_{\ell=0}^3 i^\ell \left( \|\Omega \mathbf{u} + i^\ell \Omega \mathbf{v}\|_2^2 - \|\mathbf{u} + i^\ell \mathbf{v}\|_2^2 \right) \right| \\ &\leq \frac{1}{4} \sum_{\ell=0}^3 \frac{\epsilon}{4} (\|\mathbf{u}\|_2 + \|\mathbf{v}\|_2)^2 = \epsilon \end{aligned}$$

will hold whenever (1.8) holds with  $S \leftarrow S'$  and  $\epsilon \leftarrow \epsilon/4$ . The result now follows by renormalizing.  $\square$

*Remark 3.5.* Note that if  $S \subset \mathbb{R}^n$  it suffices for a random matrix  $\Omega \in \mathbb{R}^{m \times n}$  to have the  $(\epsilon/2, \delta, 2p^2)$ -JL property for a smaller set  $S' \subset \mathbb{R}^n$  in Lemma 3.9. This can be seen by using the real version of the polarization identity instead of the complex version.

The next lemma constructs a set  $S$  to utilize in Lemma 3.9 based on two matrices with normalized columns. The end result is an entrywise approximate matrix multiplication property for the two column-normalized matrices in question.

**Lemma 3.10** (The JL property allows approximate matrix multiplies for unitary matrices). *Let  $V \in \mathbb{C}^{n \times p}$  and  $U \in \mathbb{C}^{n \times q}$  have unit  $\ell^2$ -normalized columns. Suppose that  $\Omega \in \mathbb{C}^{m \times n}$  satisfies (3.35) from Lemma 3.9 where  $S = \{\mathbf{u}_j | \mathbf{u}_j = U[:, j]\} \cup \{\mathbf{v}_j | \mathbf{v}_j = V[:, j]\}$ . Then*

$$|(V^* \Omega^* \Omega U - V^* U)_{k,j}| \leq \epsilon, \text{ for all } 1 \leq k \leq p \text{ and } 1 \leq j \leq q.$$

*Proof.* Note that  $|S| = p + q$ . Thus,  $|S'| \leq 4(p + q)^2$  in Lemma 3.9. Furthermore,

$$\Omega V = \begin{pmatrix} | & | & \dots & | \\ \Omega \mathbf{v}_1 & \Omega \mathbf{v}_2 & \dots & \Omega \mathbf{v}_p \\ | & | & \dots & | \end{pmatrix}, \text{ and } \Omega U = \begin{pmatrix} | & | & \dots & | \\ \Omega \mathbf{u}_1 & \Omega \mathbf{u}_2 & \dots & \Omega \mathbf{u}_q \\ | & | & \dots & | \end{pmatrix}.$$

Hence,  $((\Omega V)^* \Omega U)_{k,j} = \langle \Omega \mathbf{u}_j, \Omega \mathbf{v}_k \rangle$ . Therefore, given Lemma 3.9, for all choices of  $k, j$  we have

$$|(V^* \Omega^* \Omega U - V^* U)_{k,j}| = |\langle \Omega \mathbf{u}_j, \Omega \mathbf{v}_k \rangle - \langle \mathbf{u}_j, \mathbf{v}_k \rangle| \leq \epsilon \|\mathbf{v}_k\|_2 \|\mathbf{u}_j\|_2 = \epsilon.$$

$\square$

The next lemma constructs a new set  $S$  to utilize in Lemma 3.9 by selecting a well chosen subset of the singular vectors of both  $A$  and  $B$ . This set will ultimately determine how the finite set  $S$  promised by Lemma 1.8 depends on  $A$  and  $B$ . As we shall see, it's proven by applying Lemma 3.10 to two unitary matrices provided by the SVDs of  $A$  and  $B$ .

**Lemma 3.11** (The JL property implies the AMM property for arbitrary matrices). *Let  $A \in \mathbb{C}^{p \times n}$  and  $B \in \mathbb{C}^{n \times q}$  have SVDs given by  $A = U_1 \Sigma_1 V^*$  and  $B = U \Sigma_2 V_2^*$ , and suppose that  $\Omega \in \mathbb{C}^{m \times n}$  satisfies the conditions of Lemma 3.10 for  $U$  and  $V$ . Then,*

$$\|A\Omega^*\Omega B - AB\|_F \leq \epsilon \|A\|_F \|B\|_F$$

*Proof.* We will expand the quantity of interest according the SVD of  $A$  and  $B$ . Doing so we see that

$$\begin{aligned} \|A\Omega^*\Omega B - AB\|_F &= \|U_1 \Sigma_1 V^* \Omega^* \Omega U \Sigma_2 V_2^* - U_1 \Sigma_1 V^* U \Sigma_2 V_2^*\|_F \\ &= \|U_1 \Sigma_1 (V^* \Omega^* \Omega U - V^* U) \Sigma_2 V_2^*\|_F \\ &= \|\Sigma_1 (V^* \Omega^* \Omega U - V^* U) \Sigma_2\|_F \\ &= \sqrt{\sum_{k=1}^p \sum_{j=1}^q (\Sigma_1)_{k,k}^2 |V^* \Omega^* \Omega U - V^* U|_{k,j}^2 (\Sigma_2)_{j,j}^2} \\ &\leq \sqrt{\sum_{k=1}^p \sum_{j=1}^q \sigma_k(A)^2 \epsilon^2 \sigma_j(B)^2} \\ &= \epsilon \sqrt{\sum_{k=1}^p \sigma_k(A)^2} \sqrt{\sum_{j=1}^q \sigma_j(B)^2} \\ &= \epsilon \|A\|_F \|B\|_F. \end{aligned}$$

□

Lemmas 3.9, 3.10, and 3.11 now collectively prove the following generalized version of Lemma 1.8.

**Lemma 3.12** (The JL property provides the AMM property). *Let  $A \in \mathbb{C}^{p \times n}$  and  $B \in \mathbb{C}^{n \times q}$ . There exists a finite set  $S \subset \mathbb{C}^n$  with cardinality  $|S| \leq 4(p + q)^2$  (determined entirely by  $A$  and  $B$ ) such*

that the following holds: If a random matrix  $\Omega \in \mathbb{C}^{m \times n}$  has the  $(\epsilon/4, \delta, 4(p+q)^2)$ -JL property for  $S$ , then  $\Omega$  will also have the  $(\epsilon, \delta)$ -AMM property for  $A$  and  $B$ .

We will make use of this simple centering result with regards to  $L^p$  norms of random variables.

**Lemma 3.13.** *Suppose  $X$  a real random variable, and let  $p \geq 1$ , Then,*

$$\|X - \mathbb{E}[X]\|_{L^p} \leq 2 \|X\|_{L^p}$$

*Proof.* Let  $\mu = \mathbb{E}[X]$ . By Jensen's inequality,  $\|\mu\|_{L^p} = |\mu| \leq \mathbb{E}[|X|] = \|X\|_{L^1}$ . Observe,

$$\begin{aligned} \|X - \mu\|_{L^p} &\leq \|X\|_{L^p} + \|\mu\|_{L^p} \\ &\leq \|X\|_{L^p} + \|X\|_{L^1} \\ &\leq 2 \|X\|_{L^p} \end{aligned}$$

Where we have used Minkowski's inequality in the first line, and Jensen's inequality in the third.  $\square$

### 3.5.2 Proof of Theorem 3.2

A similar proof appears in support of [53, Theorem 2], which was itself simplified from earlier work [11]. We reproduce the proof here for completeness, and to clarify details. We begin by restating Theorem 3.2 for ease of reference.

**Theorem 3.12** (Restatement of Theorem 3.2). *Let  $X \in \mathbb{R}^{n \times N}$  of rank  $\tilde{r} \leq \min\{n, N\}$  have the full SVD  $X = U\Sigma V^T$ , and let  $V_{r'} \in \mathbb{R}^{N \times r'}$  denote the first  $r'$  columns of  $V \in \mathbb{R}^{N \times N}$  for all  $r' \in [N]$ . Fix  $r \in [n]$  and consider the head-tail split  $X = X_r + X_{\setminus r}$ . If  $\Omega \in \mathbb{R}^{m \times N}$  satisfies*

1. *subspace embedding property (1.10) with  $\epsilon \leftarrow \frac{\epsilon}{3}$  for  $A \leftarrow X_r^T$ ,*
2. *approximate multiplication property (1.13) with  $\epsilon \leftarrow \frac{\epsilon}{6\sqrt{\min\{r, \tilde{r}\}}}$  for  $A \leftarrow X_{\setminus r}$  and  $B \leftarrow V_{\min\{r, \tilde{r}\}}$ ,*
3. *JL property (1.8) with  $\epsilon \leftarrow \frac{\epsilon}{6}$  for  $S \leftarrow \left\{ \text{the } n \text{ columns of } X_{\setminus r}^T \right\}$ , and*
4. *approximate multiplication property (1.13) with  $\epsilon \leftarrow \frac{\epsilon}{6\sqrt{r}}$  for  $A \leftarrow X_{\setminus r}$  and  $B \leftarrow X_{\setminus r}^T$ ,*

*then  $\tilde{X} := X\Omega^T$  is an  $(\epsilon, 0, r)$ -PCP sketch of  $X$ .*

*Proof.* Let  $Q \in \mathbb{R}^{n \times r}$  be an arbitrary matrix with orthonormal columns so that  $QQ^T \in \mathbb{R}^{n \times n}$  is an orthogonal projection matrix. It suffices to show that

$$\left| \|(I - QQ^T)X\|_F^2 - \|(I - QQ^T)X\Omega^T\|_F^2 \right| \leq \epsilon \|(I - QQ^T)X\|_F^2. \quad (3.36)$$

Writing  $X$  in terms of its head-tail split, (3.36) becomes

$$\left| \|(I - QQ^T)(X_r + X_{\setminus r})\|_F^2 - \|(I - QQ^T)(X_r + X_{\setminus r})\Omega^T\|_F^2 \right| \leq \epsilon \|(I - QQ^T)X\|_F^2, \quad (3.37)$$

where  $X_r$  has the thin SVD representation  $X_r = U_r \Sigma_r V_r^T$  with  $U_r$  and  $V_r$  containing the first  $r$  columns of  $U$  and  $V$  from the full SVD of  $X$ , respectively.<sup>1</sup>

Letting  $P := (I - QQ^T)$ , and using that  $\text{tr}\{AA^T\} = \|A\|_F^2$ , one may expand the left hand side of (3.37). Noting that  $X_{\setminus r}X_r^T = 0$  and  $P = P^T$  while doing so, we can now see that

$$\begin{aligned} & \left| \|(I - QQ^T)(X_r + X_{\setminus r})\|_F^2 - \|(I - QQ^T)(X_r + X_{\setminus r})\Omega^T\|_F^2 \right| = \\ & \left| \text{tr} \left( P(X_r X_r^T + X_{\setminus r} X_{\setminus r}^T) P \right) - \text{tr} \left( P(X_r \Omega^T \Omega X_r^T + X_r \Omega^T \Omega X_{\setminus r}^T + X_{\setminus r} \Omega^T \Omega X_r^T + X_{\setminus r} \Omega^T \Omega X_{\setminus r}^T) P \right) \right|. \end{aligned}$$

Regrouping terms in the last expression while noting the invariance of trace to transposition, one can see further that

$$\begin{aligned} & \left| \text{tr} \left( P(X_r X_r^T + X_{\setminus r} X_{\setminus r}^T) P \right) - \text{tr} \left( P(X_r \Omega^T \Omega X_r^T + X_r \Omega^T \Omega X_{\setminus r}^T + X_{\setminus r} \Omega^T \Omega X_r^T + X_{\setminus r} \Omega^T \Omega X_{\setminus r}^T) P \right) \right| \\ &= \left| \text{tr} \left( P(X_r X_r^T - X_r \Omega^T \Omega X_r^T) P \right) + 2 \text{tr} \left( P(X_{\setminus r} \Omega^T \Omega X_r^T) P \right) + \text{tr} \left( P(X_{\setminus r} X_{\setminus r}^T - X_{\setminus r} \Omega^T \Omega X_{\setminus r}^T) P \right) \right| \\ &\leq \left| \text{tr} \left( P(X_r X_r^T - X_r \Omega^T \Omega X_r^T) P \right) \right| + 2 \left| \text{tr} \left( P(X_{\setminus r} \Omega^T \Omega X_r^T) P \right) \right| + \left| \text{tr} \left( P(X_{\setminus r} X_{\setminus r}^T - X_{\setminus r} \Omega^T \Omega X_{\setminus r}^T) P \right) \right|. \end{aligned}$$

Looking back at (3.37) in the light of this last computation, one can now see that it suffices to prove that

$$\begin{aligned} & \left| \text{tr} \left( P(X_r X_r^T - X_r \Omega^T \Omega X_r^T) P \right) \right| + 2 \left| \text{tr} \left( P(X_{\setminus r} \Omega^T \Omega X_r^T) P \right) \right| + \left| \text{tr} \left( P(X_{\setminus r} X_{\setminus r}^T - X_{\setminus r} \Omega^T \Omega X_{\setminus r}^T) P \right) \right| \\ &\leq \epsilon \|PX\|_F^2 \end{aligned} \quad (3.38)$$

holds to establish the desired result. Going forward we will therefore aim to prove (3.38) by proving each of the following three bounds:

$$(a) \left| \text{tr} \left( P(X_r X_r^T - X_r \Omega^T \Omega X_r^T) P \right) \right| \leq \frac{\epsilon}{3} \|PX\|_F^2,$$

---

<sup>1</sup>If  $r \geq N$  we let  $V_r = V$ .

(b)  $|\operatorname{tr}(P(X_{\setminus r}\Omega^T\Omega X_r^T)P)| \leq \frac{\epsilon}{6} \|PX\|_F^2$ , and

(c)  $|\operatorname{tr}(P(X_{\setminus r}X_{\setminus r}^T - X_{\setminus r}\Omega^T\Omega X_{\setminus r}^T)P)| \leq \frac{\epsilon}{3} \|PX\|_F^2$ .

Proving (a) – (c) will establish (3.38), thereby completing the proof.

**Proof of Bound (a):** Using again that  $\operatorname{tr}\{AA^T\} = \operatorname{tr}\{A^T A\} = \|A\|_F^2$  and that  $P = P^T$  we have

$$\left| \operatorname{tr}\left(P(X_r X_r^T - X_r \Omega^T \Omega X_r^T)P\right) \right| = \left| \|X_r^T P\|_F^2 - \|\Omega X_r^T P\|_F^2 \right|.$$

Applying Lemma 1.7 in the light of assumption 1, we now have that

$$\begin{aligned} \left| \operatorname{tr}\left(P(X_r X_r^T - X_r \Omega^T \Omega X_r^T)P\right) \right| &\leq \frac{\epsilon}{3} \|X_r^T P\|_F^2 = \frac{\epsilon}{3} \|V_r V_r^T X_r^T P\|_F^2 \\ &\leq \frac{\epsilon}{3} \|X_r^T P\|_F^2 = \frac{\epsilon}{3} \|PX\|_F^2 \end{aligned}$$

as desired.

**Proof of Bound (b):** Using the invariance of trace to both transposition and permutations, as well as that  $P^T = P = P^2$ , we can see that

$$\left| \operatorname{tr}\left(P(X_{\setminus r}\Omega^T\Omega X_r^T)P\right) \right| = \left| \operatorname{tr}\left(P(X_r\Omega^T\Omega X_{\setminus r}^T)P\right) \right| = \left| \operatorname{tr}\left(PX_r\Omega^T\Omega X_{\setminus r}^T\right) \right|.$$

Recalling the full SVD  $X = U\Sigma V^T$ , we note that if  $\tilde{r} := \operatorname{rank}(X) < \min\{n, N\}$  we can remove the last  $n - \tilde{r}$  columns of  $U$ , the last  $N - \tilde{r}$  columns of  $V$ , and the last  $n - \tilde{r}$  rows and  $N - \tilde{r}$  columns of  $\Sigma$  to form the thin SVD  $X = \tilde{U}\tilde{\Sigma}\tilde{V}^T$  with  $\tilde{U} \in \mathbb{R}^{n \times \tilde{r}}$ ,  $\tilde{\Sigma} \in \mathbb{R}^{\tilde{r} \times \tilde{r}}$ , and  $\tilde{V} \in \mathbb{R}^{N \times \tilde{r}}$ . Having done so we note that  $\tilde{\Sigma}$  will be invertable and that  $\tilde{U}\tilde{U}^T X_r = X_r$  so that we may write

$$\begin{aligned} \left| \operatorname{tr}\left(P(X_{\setminus r}\Omega^T\Omega X_r^T)P\right) \right| &= \left| \operatorname{tr}\left(PX_r\Omega^T\Omega X_{\setminus r}^T\right) \right| \\ &= \left| \operatorname{tr}\left(P\tilde{U}\tilde{U}^T X_r\Omega^T\Omega X_{\setminus r}^T\right) \right| \\ &= \left| \operatorname{tr}\left(P\tilde{U}\tilde{\Sigma}\tilde{V}^T\tilde{V}\tilde{\Sigma}^{-1}\tilde{U}^T X_r\Omega^T\Omega X_{\setminus r}^T\right) \right| \\ &= \left| \operatorname{tr}\left(\left(P\tilde{U}\tilde{\Sigma}\tilde{V}^T\right)\left(\tilde{V}\tilde{\Sigma}^{-1}\tilde{U}^T X_r\Omega^T\Omega X_{\setminus r}^T\right)\right) \right| \\ &= \left| \operatorname{tr}\left((PX)\left(\tilde{V}\tilde{\Sigma}^{-1}\tilde{U}^T X_r\Omega^T\Omega X_{\setminus r}^T\right)\right) \right|. \end{aligned}$$

Recall now that  $\langle A, B \rangle_F := \text{tr}(AB^T)$  is an inner product on matrices with  $\|A\|_F = \sqrt{\text{tr}(AA^T)}$ .

Hence, we may apply the Cauchy–Schwarz inequality to our last expression to see that

$$\begin{aligned} \left| \text{tr} \left( P(X_{\setminus r} \Omega^T \Omega X_r^T) P \right) \right| &= \left| \text{tr} \left( (PX) \left( \tilde{V} \tilde{\Sigma}^{-1} \tilde{U}^T X_r \Omega^T \Omega X_r^T \right) \right) \right| \\ &\leq \|PX\|_F \left\| \tilde{V} \tilde{\Sigma}^{-1} \tilde{U}^T X_r \Omega^T \Omega X_r^T \right\|_F \\ &= \|PX\|_F \left\| \tilde{\Sigma}^{-1} \tilde{U}^T X_r \Omega^T \Omega X_r^T \right\|_F. \end{aligned}$$

Expanding  $X_r$  in terms of its thin SVD representation  $X_r = U_r \Sigma_r V_r^T$  we now have that

$$\begin{aligned} \left| \text{tr} \left( P(X_{\setminus r} \Omega^T \Omega X_r^T) P \right) \right| &\leq \|PX\|_F \left\| \tilde{\Sigma}^{-1} \tilde{U}^T X_r \Omega^T \Omega X_r^T \right\|_F \\ &= \|PX\|_F \left\| \tilde{\Sigma}^{-1} \tilde{U}^T U_r \Sigma_r V_r^T \Omega^T \Omega X_r^T \right\|_F \\ &= \|PX\|_F \left\| V_{\min\{r, \tilde{r}\}}^T \Omega^T \Omega X_r^T \right\|_F \\ &= \|PX\|_F \left\| X_{\setminus r} \Omega^T \Omega V_{\min\{r, \tilde{r}\}} \right\|_F. \end{aligned}$$

Finally, we may now use that  $X_{\setminus r} V_{\min\{r, \tilde{r}\}} = X(I_N - V_r V_r^T) V_{\min\{r, \tilde{r}\}} = X(V_{\min\{r, \tilde{r}\}} - V_{\min\{r, \tilde{r}\}}) = 0$

to see that

$$\begin{aligned} \left| \text{tr} \left( P(X_{\setminus r} \Omega^T \Omega X_r^T) P \right) \right| &\leq \|PX\|_F \left\| X_{\setminus r} \Omega^T \Omega V_{\min\{r, \tilde{r}\}} \right\|_F \\ &= \|PX\|_F \left\| X_{\setminus r} \Omega^T \Omega V_{\min\{r, \tilde{r}\}} - X_{\setminus r} V_{\min\{r, \tilde{r}\}} \right\|_F \\ &\leq \|PX\|_F \left( \frac{\epsilon}{6\sqrt{\min\{r, \tilde{r}\}}} \|X_{\setminus r}\|_F \|V_{\min\{r, \tilde{r}\}}\|_F \right), \end{aligned}$$

where we have utilized assumption 2 in the last inequality. We are now finished after using that

$\|V_{\min\{r, \tilde{r}\}}\|_F = \sqrt{\min\{r, \tilde{r}\}}$ , and noting that  $\|X_{\setminus r}\|_F \leq \|PX\|_F$  holds for all rank  $n - r$  or greater orthogonal projections  $P$  by the definition of  $X_r$ .

**Proof of Bound (c):** Again using the invariance of trace to permutations as well as  $P = P^2 =$

$I - QQ^T$  we have that

$$\begin{aligned}
\left| \text{tr} \left( P(X_{\setminus r} X_{\setminus r}^T - X_{\setminus r} \Omega^T \Omega X_{\setminus r}^T) P \right) \right| &= \left| \text{tr} \left( P(X_{\setminus r} X_{\setminus r}^T - X_{\setminus r} \Omega^T \Omega X_{\setminus r}^T) \right) \right| \\
&= \left| \text{tr} \left( (I - QQ^T)(X_{\setminus r} X_{\setminus r}^T - X_{\setminus r} \Omega^T \Omega X_{\setminus r}^T) \right) \right| \\
&\leq \left| \text{tr} \left( X_{\setminus r} X_{\setminus r}^T - X_{\setminus r} \Omega^T \Omega X_{\setminus r}^T \right) \right| \\
&\quad + \left| \text{tr} \left( QQ^T (X_{\setminus r} X_{\setminus r}^T - X_{\setminus r} \Omega^T \Omega X_{\setminus r}^T) \right) \right| \\
&\leq \left\| X_{\setminus r}^T \right\|_F^2 - \left\| \Omega X_{\setminus r}^T \right\|_F^2 + \|QQ^T\|_F \left\| X_{\setminus r} X_{\setminus r}^T - X_{\setminus r} \Omega^T \Omega X_{\setminus r}^T \right\|_F,
\end{aligned}$$

where we have again utilized Cauchy–Schwarz in the last inequality. Utilizing assumption 3, the first term just above can be bounded by  $\frac{\epsilon}{6} \|X_{\setminus r}^T\|_F^2 = \frac{\epsilon}{6} \|X_{\setminus r}\|_F^2$  using an argument analogous to the proof of Lemma 1.7. Doing so we see that

$$\begin{aligned}
\left| \text{tr} \left( P(X_{\setminus r} X_{\setminus r}^T - X_{\setminus r} \Omega^T \Omega X_{\setminus r}^T) P \right) \right| &\leq \frac{\epsilon}{6} \|X_{\setminus r}\|_F^2 + \|QQ^T\|_F \left\| X_{\setminus r} X_{\setminus r}^T - X_{\setminus r} \Omega^T \Omega X_{\setminus r}^T \right\|_F \\
&= \frac{\epsilon}{6} \|X_{\setminus r}\|_F^2 + \sqrt{r} \left\| X_{\setminus r} X_{\setminus r}^T - X_{\setminus r} \Omega^T \Omega X_{\setminus r}^T \right\|_F.
\end{aligned}$$

Finally, we may now employ assumption 4 to bound the second term just above. Doing so we obtain that

$$\begin{aligned}
\left| \text{tr} \left( P(X_{\setminus r} X_{\setminus r}^T - X_{\setminus r} \Omega^T \Omega X_{\setminus r}^T) P \right) \right| &\leq \frac{\epsilon}{6} \|X_{\setminus r}\|_F^2 + \sqrt{r} \left\| X_{\setminus r} X_{\setminus r}^T - X_{\setminus r} \Omega^T \Omega X_{\setminus r}^T \right\|_F \\
&\leq \frac{\epsilon}{6} \|X_{\setminus r}\|_F^2 + \sqrt{r} \frac{\epsilon}{6\sqrt{r}} \|X_{\setminus r}\|_F^2 \\
&= \frac{\epsilon}{3} \|X_{\setminus r}\|_F^2.
\end{aligned}$$

To conclude we note again that  $\|X_{\setminus r}\|_F \leq \|PX\|_F$  holds for all rank  $n - r$  or greater orthogonal projections  $P$  by the definition of  $X_r$ . □

### 3.6 Algorithm Variants

There are four tasks relevant in Algorithm 3.1 where by making difference choices for these tasks, we get variants of the algorithm. Two of the tasks are related to the measurement process: sketching the tensor in order to produce leave-one-out measurements, and also sketching the tensor without leaving out a mode in order to produce measurements useful in computing the core. The second two tasks are then recovering the factor matrices, and recovering the core.

Note, if we permit a second pass on the tensor  $\mathcal{X}$  after estimating factor matrices  $Q_i$ , then core measurements are not necessary, and the optimal core given these factor matrices can be found by computing

$$\mathcal{G} = \mathcal{X} \times_1 Q_1^T \times_2 Q_2^T \cdots \times_d Q_d^T$$

See section 4.2 in [37]. This then is the computation used in two-pass versions of the algorithm; it requires first to compute the factor matrices from measurements, and then apply these factors modewise to the original tensor; no separate measurement tensor for the core is required. Crucially however, this relies on a second access to the data; for which we are computing a HOSVD.

In this section then, we break apart Algorithm 3.1 into these tasks, and show how combining different choices for these tasks produces the variants of the algorithm considered.

Within the pseudo-code, “unfold” refers to the operation of taking a tensor and flattening it into a matrix of the size listed by arranging the specified mode’s fibers as the columns. The operation “fold” is the inverse of this, taking a matrix as viewed as an unfolding along the specified mode and reshaping it into a tensor of the given dimensions.

Algorithm 3.2 takes measurement matrices (e.g. sub-Gaussian random matrices) and the data tensor  $\mathcal{X}$  and produces a set of leave-one-out measurements which can be viewed as tensors or as flattened matrices. That is, it is practical to apply the measurement matrices along the modes and obtain a tensor of measurements with  $d - 1$  modes each of length  $m$  and one mode of length  $n$ , see Figure 3.1 for a schematic depiction. The measurement process takes slices of the tensor and maps them to (smaller) slices with (mostly) shorter edge lengths. Or we can conceive of the measurements as a matrix by unfolding the measurement tensor along the mode that is uncompressed and thus obtaining a matrix of size  $n \times m^{d-1}$ , one such matrix for each mode  $i \in [d]$ . It is Kronecker structured because of the correspondence of modewise products of tensors with matrices to matrix products of unfoldings of a tensor with matrices, see for example (1.4).

Alternatively, we can use Algorithm 3.3 which also produces a set of leave-one-out measurements of the tensor  $\mathcal{X}$  which can be viewed as a matrix. It is Khatri-Rao structured because the measurement matrix applied to the unfolding is formed using Khatri-Rao products. Note, unlike

---

**Algorithm 3.2** Leave-One-Out Kronecker Sketching

---

**input :** $\Omega_{(i,j)}$  where  $\text{rank}(\Omega_{(i,i)}) = n$  and  $\Omega_{(i,j)} \in \mathbb{R}^{m \times n}$  for  $i, j \in [d], i \neq j$  $\mathcal{X}$  a  $d$  mode tensor with side lengths  $n$ **output :**  $B_i$  for  $i \in [d]$ **for**  $i \in [d]$  **do**     $\mathcal{B}_i \leftarrow \mathcal{X} \times_1 \Omega_{(i,1)} \times_2 \Omega_{(i,2)} \times_3 \cdots \times_d \Omega_{(i,d)}$     # Now unfold the measurement tensor so the mode- $i$  fibers are columns,  
    size  $n \times m^{d-1}$      $B_i \leftarrow \text{unfold}(\mathcal{B}_i, n \times m^{d-1}, \text{mode} = i)$ **end**

---

Algorithm 3.2, there is not necessarily any natural way to view the measurements  $B_i$  as tensors of  $d$  modes - the sketching process in this case takes slices of the tensor and maps them to vectors and we gather these into matrices  $B_i$  of size  $n \times m$  for each  $i \in [d]$ .

---

**Algorithm 3.3** Leave-One-Out Khatri-Rao Sketching

---

**input :** $\Omega_{(i,j)}$  where  $\text{rank}(\Omega_{(i,i)}) = n$  and  $\Omega_{(i,j)} \in \mathbb{R}^{m \times n}$  for  $i, j \in [d], i \neq j$  $\mathcal{X}$  a  $d$  mode tensor with side lengths  $n$ **output :**  $B_i$  for  $i \in [d]$ **for**  $i \in [d]$  **do**     $B_i \leftarrow \Omega_{(i,i)} X_{[i]} (\Omega_{(i,1)} \cdot \Omega_{(i,2)} \cdot \dots \cdot \Omega_{(i,i-1)} \cdot \Omega_{(i,i+1)} \cdot \dots \cdot \Omega_{(i,d)})^T$ **end**

---

In order to estimate the core, we wish another, independent set of measurements. These are obtained in the manner as Algorithm 3.2, only now we are permitted to compress all the modes, producing a tensor of  $d$  modes all with side length equal to  $m_c$ .

---

**Algorithm 3.4** Core Sketching

---

**input :** $\Phi_i \in \mathbb{R}^{m_c \times n}, i \in [d]$  $\mathcal{X}$  a  $d$  mode tensor with side lengths  $n$ **output :**  $\mathcal{B}_c$  $\mathcal{B}_c \leftarrow \mathcal{X} \times_1 \Phi_1 \times_2 \Phi_2 \times_3 \cdots \times_d \Phi_d$ 

---

Next, we describe the procedure which takes as input (a) leave-one-out measurements of  $\mathcal{X}$  (from Algorithm 3.2 or 3.3), (b) the full-rank sensing matrices applied to the uncompressed modes of  $\mathcal{X}$ , and (c) our desired target rank vector of  $\mathbf{r}$ , and then outputs a factor matrix  $Q_i$  for each mode.

In the case of exact arithmetic, no rank truncation, and no noise, this exactly recovers the factors (see (3.11)).

---

**Algorithm 3.5** Recover HOSVD Factors from Leave-One-Out Measurements

---

**input** :  
 $B_i \in \mathbb{R}^{n \times m^{d-1}}$  for  $i \in [d]$  measurements that leave mode  $i$  uncompressed  
 $\Omega_{(i,i)} \in \mathbb{R}^{n \times n}$  for  $i \in [d]$   
 $\mathbf{r} = (r, \dots, r)$  desired rank for HOSVD  
**output** :  $Q_1, \dots, Q_d \in \mathbb{R}^{n \times r}$   
# Factor matrix recovery  
**for**  $i \in [d]$  **do**  
| # Solve  $n \times n$  linear system  
| Solve  $\Omega_{(i,i)} F_i = B_i$  for  $F_i$   
| # Compute SVD and keep the top  $r$  singular vectors  
|  $U, \Sigma, V^T \leftarrow \text{SVD}(F_i)$   
|  $Q_i \leftarrow U_{:,r}$   
**end**

---

Lastly, we consider the task of obtaining the core of the HOSVD of the data tensor. The two ways described in section 3.2 are to either compute the core using a second access to the data tensor - in which case this is a matter of applying the transpose of the factor matrices from Algorithm 3.5 to the data tensor. This is detailed in Algorithm 3.10.

---

**Algorithm 3.6** Compute HOSVD Core with Second Access

---

**input** :  
 $Q_1, \dots, Q_d \in \mathbb{R}^{n \times r}$  computed factor matrices  
 $\mathcal{X}$  a  $d$  mode tensor with side lengths  $n$ .  
**output** :  $\mathcal{G}$  a  $d$  mode tensor with side lengths  $r$   
 $\mathcal{G} = \mathcal{X} \times_1 Q_1^T \times_2 Q_2^T \cdots \times_d Q_d^T$

---

In the scenario in which a second access to the tensor is not desired, instead we obtain the core of the HOSVD by solving a linear system involving the measurement operators and the factor matrices, see (3.21). This is equivalent to solving the linear system a mode at a time as detailed in Algorithm 3.7, a method of practical value because it does not require as much working memory.

A third possibility is to “re-use” leave-one-out measurements to compute the core. Theoretically this involves new dependencies on the errors introduced by estimating the factors and errors introduced by recovering the core that are not addressed in any of our main results. Practically

---

**Algorithm 3.7** Recover HOSVD Core from Measurements

---

**input :** $\mathcal{B}_c$  a  $d$  mode tensor with side lengths  $m_c$  $\Phi_i \in \mathbb{R}^{m_c \times n}$  $Q_1, \dots, Q_d \in \mathbb{R}^{n \times r}$  computed factor matrices**output :**  $\mathcal{H}$ **for**  $i \in [d]$  **do**# unfold measurements, mode- $i$  fibers are columns, size  $m_c \times r^{(i-1)} m_c^{d-1-(i-1)}$  $H \leftarrow \text{unfold}(\mathcal{B}_c, m_c \times r^{(i-1)} m_c^{d-1-(i-1)}, \text{mode} = i)$ # Undo the mode- $i$  measurement operator and factor's action by finding least square solution to  $m_c \times r$  over-determined linear systemSolve  $\Phi_i Q_i H_{\text{new}} = H$  for  $H_{\text{new}}$ 

# reshape the flattened partially solved core into a tensor

 $\mathcal{B}_c \leftarrow \text{fold}(H_{\text{new}}, \underbrace{r \times r \cdots \times r}_i \times \underbrace{m_c \times \cdots \times m_c}_{d-i}, \text{mode} = i)$ # Each iteration  $m_c \rightarrow r$  in  $i$ th mode**end**

---

however, it would be desirable to avoid having to produce the core measurement tensor, and empirically on synthetic data the overall error is not effected.

---

**Algorithm 3.8** Recover HOSVD Core from Recycled Leave One Out Measurements

---

**input :** $B_j$  for a fixed  $j$ . $\Omega_{j,i}$  for each  $i \in [d]$  $Q_1, \dots, Q_d \in \mathbb{R}^{n \times r}$  computed factor matrices**output :**  $\mathcal{H}$ **for**  $i \in [d]$  **do**# unfold measurements, mode- $i$  fibers are columns, size  $m \times r^{(i-1)} m^{d-1-(i-1)}$  $H \leftarrow \text{unfold}(B_j, m \times r^{(i-1)} m^{d-1-(i-1)}, \text{mode} = i)$ # Undo the mode- $i$  measurement operator and factor's action by finding least square solution to  $m \times r$  over-determined linear systemSolve  $\Omega_{(j,i)} Q_i H_{\text{new}} = H$  for  $H_{\text{new}}$ 

# reshape the flattened partially solved core into a tensor

 $B_j \leftarrow \text{fold}(H_{\text{new}}, \underbrace{r \times r \cdots \times r}_i \times \underbrace{m_c \times \cdots \times m_c}_{d-i}, \text{mode} = i)$ # Each iteration  $m \rightarrow r$  in  $i$ th mode or  $n \rightarrow n$  when  $i = j$ **end**

---

We are now able to state the variants of the general algorithm.

---

**Algorithm 3.9** Recover HOSVD Kronecker One Pass

---

# Obtain measurements for factors with Alg. 3.2  
 $B_1, B_2, \dots, B_d \leftarrow \text{Leave-One-Out Kronecker Sketching}(\{\Omega_{(i,j)}\}_{i,j \in [d]}, \mathcal{X})$   
# Obtain measurement for core with Alg. 3.4  
 $\mathcal{B}_c \leftarrow \text{Core Sketching}(\{\Phi_i\}_{i \in [d]}, \mathcal{X})$   
# Estimate factor matrices using Alg. 3.5  
 $Q_1, Q_2, \dots, Q_d \leftarrow \text{Recover HOSVD Factors from}$   
 $\text{Leave-One-Out Measurements}(\{\Omega_{(i,i)}\}_{i \in [d]}, \{B_i\}_{i \in [d]}, (r, \dots, r))$   
# Estimate core using Alg. 3.8  
 $\mathcal{H} \leftarrow \text{Recover HOSVD Core from Measurements}(\{\Phi_i\}_{i \in [d]}, \mathcal{B}_c)$   
**output** :  $\hat{\mathcal{X}} = \llbracket \mathcal{H}, Q_1, \dots, Q_d \rrbracket$

---

---

**Algorithm 3.10** Recover HOSVD Kronecker Two Pass

---

# Obtain measurements for factors with Alg. 3.2  
 $B_1, B_2, \dots, B_d \leftarrow \text{Leave-One-Out Kronecker Sketching}(\{\Omega_{(i,j)}\}_{i,j \in [d]}, \mathcal{X})$   
# Estimate factor matrices using Alg. 3.5  
 $Q_1, Q_2, \dots, Q_d \leftarrow \text{Recover HOSVD Factors from}$   
 $\text{Leave-One-Out Measurements}(\{\Omega_{(i,i)}\}_{i \in [d]}, \{B_i\}_{i \in [d]}, (r, \dots, r))$   
# Compute core using Alg. 3.6  
 $\mathcal{H} \leftarrow \text{Compute HOSVD Core with Second Access}(\{Q_i\}_{i \in [d]}, \mathcal{X})$   
**output** :  $\hat{\mathcal{X}} = \llbracket \mathcal{H}, Q_1, \dots, Q_d \rrbracket$

---

---

**Algorithm 3.11** Recover HOSVD Khatri-Rao One Pass

---

# Obtain measurements for factors with Alg. 3.3  
 $B_1, B_2, \dots, B_d \leftarrow \text{Leave-One-Out Khatri-Rao Sketching}(\{\Omega_{(i,j)}\}_{i,j \in [d]}, \mathcal{X})$   
# Obtain measurements for core with Alg. 3.4  
 $\mathcal{B}_c \leftarrow \text{Core Sketching}(\{\Phi_i\}_{i \in [d]}, \mathcal{X})$   
# Estimate factor matrices using Alg. 3.5  
 $Q_1, Q_2, \dots, Q_d \leftarrow \text{Recover HOSVD Factors from Leave-One-Out Measurements}$   
 $(\{\Omega_{(i,i)}\}_{i \in [d]}, \{B_i\}_{i \in [d]}, (r, \dots, r))$   
# Estimate core using Alg. 3.8  
 $\mathcal{H} \leftarrow \text{Recover HOSVD Core from Measurements}(\{\Phi_i\}_{i \in [d]}, \mathcal{B}_c)$   
**output** :  $\hat{\mathcal{X}} = \llbracket \mathcal{H}, Q_1, \dots, Q_d \rrbracket$

---

---

**Algorithm 3.12** Recover HOSVD Khatri-Rao Two Pass

---

# Obtain measurements for factors with Alg. 3.3

$B_1, B_2, \dots, B_d \leftarrow \text{Leave-One-Out Khatri-Rao Sketching}(\{\Omega_{(i,j)}\}_{i,j \in [d]}, \mathcal{X})$

# Estimate factor matrices using Alg. 3.5

$Q_1, Q_2, \dots, Q_d \leftarrow \text{Recover HOSVD Factors from Leave-One-Out Measurements}$   
 $(\{\Omega_{(i,i)}\}_{i \in [d]}, \{B_i\}_{i \in [d]}, (r, \dots, r))$

# Compute core using Alg. 3.6

$\mathcal{H} \leftarrow \text{Compute HOSVD Core with Second Access}(\{Q_i\}_{i \in [d]}, \mathcal{X})$

**output :**  $\hat{\mathcal{X}} = \llbracket \mathcal{H}, Q_1, \dots, Q_d \rrbracket$

---

## CHAPTER 4

### TENSOR SANDWICH: TENSOR COMPLETION FOR LOW CP-RANK TENSORS VIA ADAPTIVE RANDOM SAMPLING

In the proceeding chapters we have described two schemes for the tensor recovery problem. We now will address a method for a different, though related, tensor problem. In tensor *completion* we consider the problem of how we might complete (or impute, or predict) a tensor from which some entries are missing or unavailable. That is, for a tensor  $\mathcal{X} \in \mathbb{R}^{n_1 \times \dots \times n_d}$ , we have some set,  $\Omega \subset [n_1] \times \dots \times [n_d]$  of indices which correspond to entries which are sampled, or revealed. Entries with indices outside of the revealed set are then not available and some or all of these missing entries we desire to fill in, and do so in a way which minimizes error from the true tensor.

Naturally, if the entries are unrelated to each other, the problem is hopeless, however in the case where the tensor has additional structure such as a low-rank decomposition, then we can use that fact to develop methods which are able to accurately compute (or estimate in the inexact case) given only a small sample of some of the entries. The commonalities with the recovery problem are immediate - the sampling operator can fruitfully be envisioned as linear measurements of the original tensor and from those measurements we wish to recover the tensor, or a good estimate for it. It is often the case that we wish to limit or minimize how many entries are revealed. However, we will quickly leave behind the requirement that this operator act in a modewise manner or with interleaved reshapings as was done in the previous chapters. One reason to not insist on an operator conforming to such a scheme is that we will want to discuss sampling schemes which are *adaptive*, in the sense that after observing some entries, we will use some computation and information gleaned from these entries to update where next we would like to sample in order to best complete the tensor. In this case the set  $\Omega$  is constructed in stages. The degree to which a sampling scheme adapts or not, and the reasonableness of having different levels of control over the sampling pattern is something we shall discuss.

## 4.1 Adaptive Tensor Sandwich for Completing Three Mode Tensors

The subject of this chapter, and our proposed method for addressing the tensor completion problem resulted from combining *matrix* completion techniques (see, e.g., [8, 40, 10]) for a small number of slices (see Definition 1.3) with a modified noise robust version of Jennrich’s algorithm (see, e.g., [51, Section 3]); an algorithm which originally was used to prove (constructively) the uniqueness of the CP decomposition when rank is less than or equal to the smallest side length of the tensor.

In the simplest case as we shall see, this leads to a sampling strategy that more densely samples two slices (the bread), and then more sparsely samples additional entries throughout the tensor (the filling) for final completion, hence “Tensor Sandwich”. To begin, we will consider three mode tensors with some mild assumptions on the factor matrices of the CP decomposition (see Definition 1.14). Our algorithm then completes an  $n \times n \times n$  tensor with CP-rank  $r$  with high probability while using at most  $O(nr \log^2 r)$  adaptively chosen samples.

Empirical experiments further verify that the approach works well in practice, including in the presence of additive noise, see Section 4.3. We will then include details for how the method can be employed for tensors of more than three modes and how sampling can be modified to be essentially non-adaptive, for the cases in which a practitioner may not have the level of control needed to conform to the adaptive strategy, see Section 4.5.

Before beginning our description of our proposed adaptive method, we make two important observations. First, there is a simple information theoretic lower bound on any completion method, adaptive or otherwise. For example, a  $d$  mode CP-rank  $r$  tensor, absent any other constraint, has  $rd(n - 1) + r$  degrees of freedom, and so no sampling scheme which uses fewer entries than this can guarantee an exact completion. Furthermore, any method which employs randomness should then require more samples, and though we do not have a proof that our method is strictly optimal in this regard, it is by comparison to other known methods, better, and does have scaling up to log factors in the relevant parameters of the problem.

Second, there is not *a priori* any guarantee that a particular revealed entry in a tensor will be

especially informative. For example, consider the rank one tensor  $\mathcal{T} = \bigcirc_{j=1}^d \mathbf{e}_1$  where  $\mathbf{e}_1$  is the standard basis vector with a one in position one. All but one of this tensors'  $n^d$  entries are zero, which is to say, a sampling technique would have to reveal this singular non-zero entry in order to complete the tensor and - any random sampling pattern would need on the order the number of entries of the tensor revealed to succeed in such a case. Fortunately such “worst” case tensors can be excluded with some natural and useful assumptions.

#### 4.1.1 Statement and Outline of Main Result

To proceed with defining the class of tensors for which the method will have a high chance of completing, we will need two additional definitions, motivated by the preceding discussion of the completion problem.

**Definition 4.1** (Coherence). The *coherence* of an  $r$ -dimensional subspace  $U \subset \mathbb{R}^n$  is defined

$$\mu(U) := \frac{n}{r} \max_{i \in [n]} \|\mathcal{P}_U \mathbf{e}_i\|_2^2,$$

where  $\mathcal{P}_U$  is the orthogonal projection onto  $U$ , and where  $\mathbf{e}_i \in \mathbb{R}^n$  for  $i \in [n]$  denotes the  $i^{\text{th}}$  standard basis vector.

From this definition we can see a subspace will have low coherence or be *incoherent* if, in terms of magnitude and support, its entries are (roughly) equal and spread out among all possible indices. In such a case, it will happen that revealing an entry has a high likelihood of being informative - the concentration of mass will not be constrained to some relatively small support of the total index set, so the chance that a sampling scheme “misses” informative entries is lower.

We will also need the following definition concerning an elaboration on rank of a matrix.

**Definition 4.2** (Kruskal rank). The *Kruskal rank* of a matrix is the maximum integer  $r$  such that any  $r$  columns of the matrix are linearly independent.

We observe immediately that Kruskal rank is always bounded by the rank of the matrix. As a simple yet illuminating example, consider the identity matrix in  $\mathbb{R}^{n \times n}$ , which has rank and Kruskal rank both equal to  $n$ . Now append a column to this which consists of  $\mathbf{e}_1$ . Clearly the rank of the resulting matrix remains  $r$ , but the Kruskal rank is now one, since it is possible to find a subset of

two columns among the  $n + 1$  columns which are linearly dependent.

The class of tensors for which the proposed approach will be guaranteed to succeed will need to satisfy the following assumptions. For positive integers  $n, r, s$  with  $r, s \leq n$  and any  $\mu_0 \in [1, n/r]$ , we define  $\mathbb{T}(n, r, \mu_0, s)$  to be the class of all tensors  $\mathcal{T} = \sum_{i=1}^r \mathbf{a}_i \circ \mathbf{b}_i \circ \mathbf{c}_i \in \mathbb{R}^{n \times n \times n}$  for which:

- (a) the factor matrices  $A, B$  both have full column rank,
- (b) the column space of  $A$  has coherence bounded above by  $\mu_0$ , and
- (c) for some  $s \in [n]$ , every  $s \times r$  submatrix of  $C$  has Kruskal rank  $\geq 2$ .

We are now able to state our main result of this section.

**Theorem 4.1.** *There exists an adaptive random sampling strategy and an associated reconstruction algorithm (see Algorithm 4.2) such that for any  $\delta > 0$ , after observing at most  $C_1 s \mu_0 n r \log^2(r^2/\delta)$  entries of a tensor  $\mathcal{T} \in \mathbb{T}(n, r, \mu_0, s)$ , the algorithm completes  $\mathcal{T}$  with probability at least  $1 - \delta$ . Here  $C_1 > 0$  is absolute constant that is independent of all other quantities.*

The proof of our sandwich sampling algorithm involves three stages. First, without loss of generality, we pick  $s$  mode-3 slices and then use a matrix completion algorithm that with high probability recovers these slices and observes at most  $C'_1 \mu_0 n r \log^2(r^2/\delta)$  entries in each slice. Second, if this  $n \times n \times s$  sub-tensor is correctly completed, we can then use a deterministic variant of Jennrich's Algorithm<sup>1</sup> on the completed sub-tensor to learn the factor matrices  $A$  and  $B$ . Third, once we know the factor matrices  $A$  and  $B$ , we can deterministically find  $r$  sample locations in each of the  $n$  mode-3 slices whose values allow a censored least squares problem to solve for the third factor matrix  $C$ . This three-stage procedure uses at most  $C'_1 s \mu_0 n r \log^2(r^2/\delta)$  samples to complete the  $s$  initial mode-3 slices in order to learn  $A$  and  $B$ , and then  $nr$  additional samples to learn  $C$  thereafter, for a total of at most  $C'_1 s \mu_0 n r \log^2(r^2/\delta) + nr \leq C_1 s \mu_0 n r \log^2(r^2/\delta)$  samples. See Figure 4.1 for a schematic illustration of the overall sampling strategy where fibers are sampled through the middle of the sandwich for simplicity.

We note that assumptions (a) and (c) are the necessary assumptions for the Jennrich's step to work with any  $n \times n \times s$  sub-tensor. Furthermore, if the columns of the factor matrices are

---

<sup>1</sup>As discussed in [36] the work of [44] is perhaps a more accurate attribution of the method, however we use the traditional name of Jennrich's Algorithm in this work.

drawn from any continuous distribution, assumption (c) for  $s = 2$  holds with probability 1. Finally, something akin to assumption (b) is always required in completion problems, as we have discussed in Section 4.1.

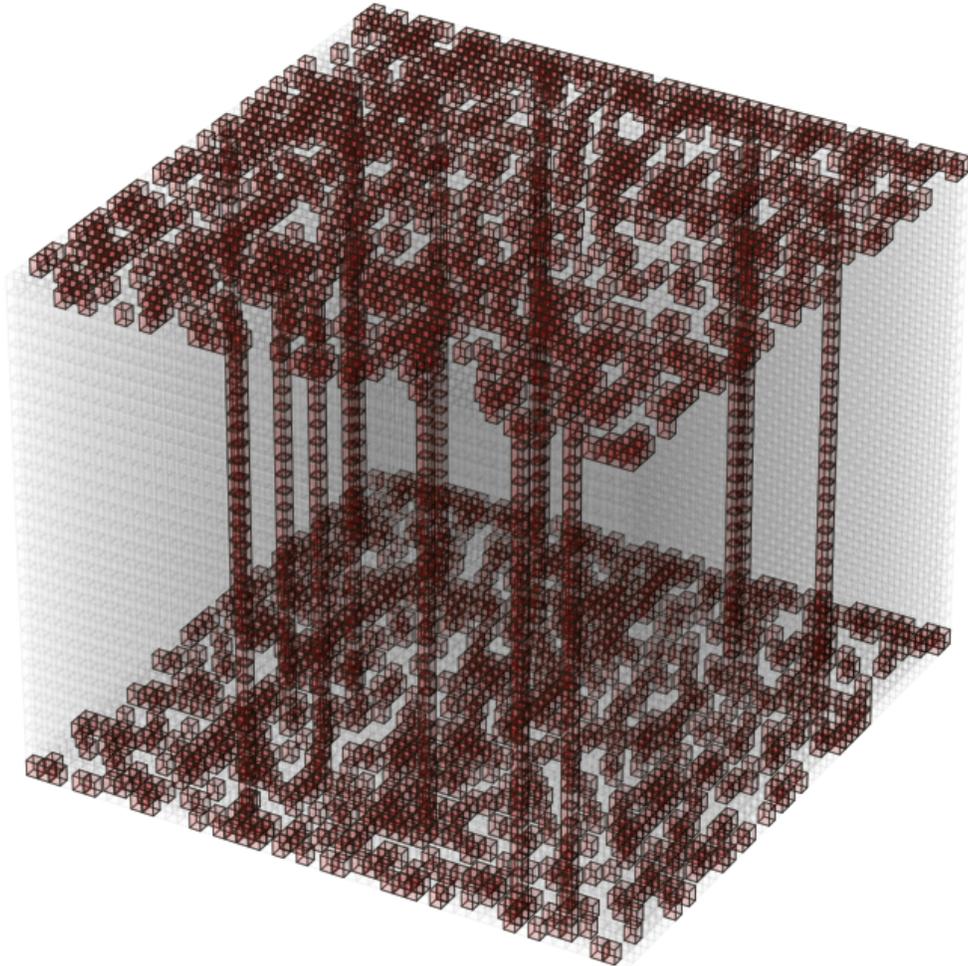


Figure 4.1 A schematic depiction of the sampling strategy where  $s = 2$  slices have been sampled relatively densely in order to compute  $A$  and  $B$ , and where additional fibers were then sampled elsewhere to use in computing  $C$ .

#### 4.1.2 Related Work

Many prior works on low-rank tensor completion use non-adaptive and uniform sampling [29, 4, 65, 66, 55, 52, 45, 34]. While some of those works [4, 34, 52] can handle CP-ranks up to roughly  $n^{3/2}$  instead of  $n$ , all of them require at least  $O(n^{3/2})$  samples, even when the rank is  $r = O(1)$ . Furthermore, [4] shows that completing an  $n \times n \times n$  rank- $r$  tensor from  $n^{3/2-\epsilon}$  uniformly

random samples is NP-hard by way of showing it is equivalent to the problem of refuting a 3-SAT formula with  $n$  variables and  $n^{3/2-\epsilon}$  clauses.

In [39], the authors propose a recursive method for completing CP-rank  $r \leq n$  tensors using adaptive sampling which, for order-3 tensors, requires  $O(\mu_0^2 nr^{5/2} \log^2 r)$  samples. Our algorithm requires a number of samples which has a more favorable dependence on the coherence  $\mu_0$  and rank  $r$ . Furthermore, our result only requires coherence assumptions about  $A$ , instead of about  $A$  and  $B$ . These improvements come at the expense of requiring the mild additional assumptions (a) and (c) in our Theorem 4.1 related to Jennrich’s algorithm, however. Additionally, the method of [39], outside the exact case does not output a CP factorization of the tensor, but rather a more complicated and recursive decomposition.

The first step in our tensor completion algorithm involves using an adaptive sampling scheme to complete  $s$  mode-3 slices of the tensor. Our tensor completion algorithm and results are based on the adaptive matrix completion algorithm and results in [40] which, with high probability, uses  $O(\mu_0 nr \log^2 r)$  samples to complete a rank  $r$  matrix. However, our algorithm can be adapted to use other adaptive matrix completion results such as, e.g., [10] with relative ease.

In the censored least squares phase of our algorithm, we can sample entire fibers of the tensor as done in Figure 4.1. We note that doing so is similar in spirit to the fiber sampling approach of Sørensen and De Lathauwer [58]. However, their work focuses on determining algebraic constraints on the factor matrices of a low rank tensor which, when satisfied, allow the tensor to be completed from the sampled fibers. As such, our results cannot be directly compared with [58].

## 4.2 Proof of Tensor Sandwich Exact Recovery for Tensors in $\mathbb{T}(n, r, \mu_0, s)$

In this section we follow our three stage proof outline from Section 4.1.1

### 4.2.1 Completing $s$ Mode-3 Slices of $\mathcal{T}$

We start by picking any subset of indices  $S \subset [n]$  with  $|S| = s$  elements. For each  $k \in S$ , the mode-3 slice  $\mathcal{T}_{:, :, k} \in \mathbb{R}^{n \times n}$  satisfies

$$\mathcal{T}_{:, :, k} = \sum_{i=1}^r \langle \mathbf{c}_i, \mathbf{e}_k \rangle \mathbf{a}_i \mathbf{b}_i^T,$$

and so,  $\text{col-span}(\mathcal{T}_{:,k}) \subseteq \text{col-span}(A)$ . By assumption,  $\text{col-span}(A)$  has coherence bounded by  $\mu_0$ . Thus, the assumptions required by Theorem 1 in [40] hold. Therefore, for each slice  $T_{:,k}$   $k \in S$ , with probability at least  $1 - \delta$ , the adaptive sampling procedure in Algorithm 1 uses at most  $C'_1 \mu_0 n r \log^2(r^2/\delta)$  samples for some absolute constant  $C'_1 > 0$ , and completes  $\mathcal{T}_{:,k}$ .

By taking a simple union bound over each of the  $s$  slices  $k \in S$ , we have that with probability at least  $1 - s\delta$ , this strategy will successfully complete all of the  $s$  slices  $\mathcal{T}_{:,k}$  for  $k \in S$  and use fewer than  $C'_1 s \mu_0 n r \log^2(r^2/\delta)$  samples. Let  $\Omega_1 = \{(i, j, k) | (i, j) \text{ sampled according to [40] for slice } k \in S\}$ , the set of locations of  $\mathcal{T}$  sampled to complete these  $s$  slices.

#### 4.2.2 Learning Mode-1 and 2 Factor Matrices via Modified Jennrich's Algorithm / Simultaneous Diagonalization

Let  $\mathbf{u}, \mathbf{v}$  be random vectors uniformly drawn from the unit sphere  $\mathbb{S}^{s-1}$ . Denote the sub-vector of  $\mathbf{c}$  with entries indexed in  $S$  by  $\tilde{\mathbf{c}}_i = (\mathbf{c}_i)_S$ , and construct two auxiliary matrices  $T_u, T_v$  using the completed slices by adding up the linear combinations of the completed slices, weighted by the random vectors  $\mathbf{u}, \mathbf{v}$ . In terms of the components, we have:

$$T_u = \sum_{i=1}^r \langle \tilde{\mathbf{c}}_i, \mathbf{u} \rangle \mathbf{a}_i \mathbf{b}_i^T$$

$$T_v = \sum_{i=1}^r \langle \tilde{\mathbf{c}}_i, \mathbf{v} \rangle \mathbf{a}_i \mathbf{b}_i^T$$

Denote the  $r \times r$  matrix  $D_u$  which has along its diagonal entries the values  $\langle \tilde{\mathbf{c}}_i, \mathbf{u} \rangle$ , and similarly  $D_v$ . Notice the following identity with the product  $T_u(T_v)^\dagger$

$$\begin{aligned} T_u(T_v)^\dagger &= A D_u B^T (A D_v B^T)^\dagger \\ &= A D_u D_v^{-1} A^\dagger \end{aligned}$$

Where the matrix  $D_u D_v^{-1}$  is a diagonal matrix with  $(i, i)$ -entry equal to  $\frac{\langle \tilde{\mathbf{c}}_i, \mathbf{u} \rangle}{\langle \tilde{\mathbf{c}}_i, \mathbf{v} \rangle}$ . Clearly the matrix  $T_u(T_v)^\dagger$  is diagonalizable, and so by computing the eigen-decomposition of  $T_u(T_v)^\dagger$  we recover the columns of  $A$  and the eigenvalues along the diagonal of  $D_u D_v^{-1}$ . We can order the eigenvectors in descending order by magnitude of their corresponding eigenvalue. Note it is here that we employ

assumption (c): Since  $C_S$  (i.e., the rows of  $C$  indexed by  $S$ ) has  $k$ -rank at least 2, the ratios  $\frac{\langle \tilde{\mathbf{c}}_i, \mathbf{u} \rangle}{\langle \tilde{\mathbf{c}}_i, \mathbf{v} \rangle}$  for each  $i$  will almost surely be distinct from one another, and thus the ordering of the eigenvalues is unique.

Let  $P$  be the permutation matrix that interchanges the columns of  $A$  so that instead of being in  $D_u D_v^{-1}$  order, they are in  $D_u$  order (ordered greatest to least in terms of the magnitude of  $\langle \tilde{\mathbf{c}}_i, \mathbf{u} \rangle$ ). Now notice that

$$\begin{aligned} A^\dagger T_u &= A^\dagger A P D_u B^T \\ &= P D_u B^T \\ &= (B D_u P)^T. \end{aligned} \tag{4.1}$$

This means that the re-scaled columns of  $B$  that are in  $D_u$  order are now in  $D_u D_v^{-1}$  order after we apply the inverse  $A^\dagger$  to  $T_u$ . That is, we have found the matching components of  $A$  and  $B$  up to a re-scaling of their outer product! This was achieved by learning the columns of  $A$  from  $T_u (T_v)^\dagger$  and then, crucially, the matching columns of  $B$  from  $A^\dagger T_u$ . The scaling due to  $D_u$  will be resolved in the final step of the algorithm, where we solve for the missing third factor (i.e.,  $B, C$  and  $BD, C(D^{-1})$  are both valid pairs of factors for any diagonal matrix  $D$ ).

### 4.2.3 Learning the mode-3 factor matrix

After obtaining  $A$  and a re-scaled  $B$  in order to find the remaining components,  $C$ , we will need  $\Omega_2$ , a second set of revealed locations of the entries of  $\mathcal{T}$ . We will also need the solutions to  $n$  instances of the following censored least squares problem related to those revealed values

$$(A \odot B)_{K_k} \mathbf{c} = \mathbf{t}_{K_k} \tag{4.2}$$

where  $k \in [n], \mathbf{c} = (C_{k,:})^T, \mathbf{t} = \text{vec}(\mathcal{T}_{:, :, k})^T$  and  $K_k = \{i + n(j - 1) \mid (i, j, k) \in \Omega_2\}$ . In (4.2), the term  $\mathbf{t}_{K_k}$  denotes the vector of length  $|K_k|$  which includes only the entries of  $\mathbf{t}$  which have indices appearing in the set  $K_k$ . Similarly  $(A \odot B)_{K_k}$  is the matrix where we restrict rows of  $(A \odot B)$  to only those which have indices appearing in the set.

Note  $(A \odot B)$  is full rank because  $A$  and  $B$  are assumed to be full rank (see, e.g., [31]). This implies the uncensored system is consistent with a unique solution. The difficulty in the censored

case is that unobserved values in a particular column of the right hand side of (4.2) could force the discarding of rows of the matrix  $(A \odot B)$  that cause the system to become under-determined.

That is, we must ensure that  $(A \odot B)_{K_k}$  has full column rank for each  $k$ . If we do not vary our sampling procedure from frontal slice to frontal slice, then this corresponds to sampling tubes of the original tensor of the form  $\mathcal{T}_{i,j,:}$  and  $K_k = K$  for all  $k \in [n]$ . In order to arrange the sampling to accomplish this, consider a  $QR$  with column-pivoting factorization of  $(A \odot B)^T$ , see algorithm 5.4.1 in [19]). This produces factors and a permutation matrix  $\mathbb{C}^{n_1 \times \dots \times n_d} \in \mathbb{R}^{n^2 \times n^2}$  such that  $(A \odot B)^T \mathbb{C}^{n_1 \times \dots \times n_d} = QR$ , where

$$R = \begin{bmatrix} R_1 & R_2 \end{bmatrix}$$

and  $R_1 \in \mathbb{R}^{r \times r}$  is upper-triangular and non-singular. But this means that the first  $r$  columns of  $\mathbb{C}^{n_1 \times \dots \times n_d}$  select a set of columns of  $(A \odot B)^T$  which are linearly independent, so for each  $\mathbb{C}_{:,p}^{n_1 \times \dots \times n_d}$ ,  $p \in [r]$  we have that there exists some  $q \in [n^2]$  such that  $\mathbf{e}_q = \mathbb{C}_{:,p}^{n_1 \times \dots \times n_d}$ , the corresponding column of the identity in  $\mathbb{R}^{n^2 \times n^2}$ . Let  $i = q \bmod n$ ,  $j = \lfloor \frac{q}{n} \rfloor$ . Define then  $\Omega_2$  all the tuples of the form  $(i, j, :)$ . That is, we can read off from  $\mathbb{C}^{n_1 \times \dots \times n_d}$  which  $r$  fibers of length  $n$  to sample in order to ensure (4.2) is consistent for each  $k \in [n]$ . We have specified at most  $nr$  new sample locations, and thus  $\Omega = \Omega_1 \cup \Omega_2$ , the set of all samples employed to complete the tensor  $\mathcal{T}$  is at most  $|\Omega_1| + |\Omega_2| \leq C'_1 s \mu_0 nr \log^2(r^2/\delta) + nr \leq C_1 s \mu_0 nr \log^2(r^2/\delta)$ .

*Remark 4.1.* Note,  $\mathbb{C}^{n_1 \times \dots \times n_d}$  is not unique, and indeed we could use different selections corresponding to other valid permutations from frontal slice to frontal slice to vary the sampling pattern when finding  $C$ . Additionally, including more rows of  $(A \odot B)$  beyond the first  $r$  as specified by  $\mathbb{C}^{n_1 \times \dots \times n_d}$  may be numerically advantageous when computing  $C$ .

### 4.3 Experiments

In this section, we show that tensors sampled according to our overall strategy and completed using Algorithm (4.2) support our theoretical findings. In particular we demonstrate that once sample complexity bounds are satisfied, we can achieve very precise levels of relative error by sampling what is overall a small percentage of the total tensor's entries. For example, in the rank 20

---

**Algorithm 4.1** Consistency Preserving Fiber Sampler

---

**input** : $\gamma \geq 1$ , over-sample parameter $r$  rank parameter $A, B \in \mathbb{R}^{n \times r}$ , estimates for factor matrices**output** :  $\Omega_2$ , set of sample locationsCompute  $QR$  with column-pivoting,  $QR = (A \odot B)^T \mathbb{C}^{n_1 \times \dots \times n_d}$ **for**  $p \in [\gamma r]$  **do**     $q \leftarrow \text{nonzero}(\mathbb{C}_{:,p}^{n_1 \times \dots \times n_d})$      $i \leftarrow q \bmod n$      $j \leftarrow \lfloor \frac{q}{n} \rfloor$     Include  $(i, j, k)$  in  $\Omega_2$  for all  $k \in [n]$ **end**

---

---

**Algorithm 4.2** Tensor Sandwich

---

**input** :  $S \subseteq [n]$  slices to complete**output** :  $\hat{T}$ , completed tensor

# Slice Complete Phase

**for**  $k \in S$  **do**    Use, e.g., [10], or algorithm 1 in [40] to complete  $\mathcal{T}_{:, :, k}$ . Algorithm 1 in [40] uses at most  $C_1 \mu_0 n r \log^2(r^2/\delta)$  adaptively chosen samples.**end**

# Jennrich Complete Phase

Generate random vectors  $\mathbf{u}, \mathbf{v} \in \mathbb{S}^{s-1}$ ; $T_u \leftarrow \sum_{k \in S} u_k T_{:, :, k}$  $T_v \leftarrow \sum_{k \in S} v_k T_{:, :, k}$ Compute eigen-decomposition  $A \Lambda A^{-1} = T_u (T_u)^\dagger$  $B \leftarrow A^{-1} T_u$ 

# Censored Least Squares Phase

 $\Omega_2 \leftarrow \text{fiber sampler}(A, B, r, \gamma)$  $C^T \leftarrow \text{censored least squares}((A \odot B), \mathcal{T}_{i,j,k} \text{ s.t. } (i, j, k) \in \Omega_2)$  $\hat{T} \leftarrow [[A, B, C]]$ 

---

case in Figure 4.2, the median relative error is 0.000168 having sampled  $\frac{94445}{8000000} \approx 1.1\%$  of the total entries. Our experiments also verify the linear dependence on rank in the sample complexity bound. Furthermore, in our simulated data and real world data we show empirical evidence the method performs useful completion even in the presence of noise. Additionally we make comparisons to other tensor completion methods.

### 4.3.1 Simulated Data

In this section, we show that Tensor Sandwich, as well as its extension to higher orders (see Section 4.5), which we refer to as Tensor Deli in the figures, can complete tensors of simulated data using both adaptive and non-adaptive sampling. Once sample complexity bounds are satisfied, we can achieve precise levels of relative error for completing a tensor by looking at what is overall a small fraction of its total entries. We also demonstrate empirically our method works for four mode tensors. In prior works such as [39], results of this kind are given only for two mode tensors; in [45] tensors of three modes. Furthermore we show how using Tensor Sandwich/Deli to initialize a masked-alternating least squares scheme with even a few iterations provides significant improvement in accuracy. We compare our adaptive results to [39] and [45], and show that Tensor Deli can perform well in terms of accuracy and sample complexity while also providing a factorized tensor.

In all cases with the simulated data, the tensor is generated by drawing factor matrices of shape  $\mathbb{R}^{n \times r}$  with i.i.d standard Gaussian entries and then normalizing the columns. For simulated data, for each mode  $d$ , the side-lengths  $n$  are the same, and thus the tensor has  $n^d$  total entries. We then have the option to weight the components using decaying weights described by the parameter  $\alpha$ , i.e.

$$\mathcal{T} = \sum_{\ell=1}^r \left( \frac{1}{\ell^\alpha} \right) \mathbf{a}_\ell^{(1)} \circ \dots \circ \mathbf{a}_\ell^{(d)} \quad (4.3)$$

In each experiment, the median of the errors is taken over ten independent trials. In each Tensor Deli experiment, after selecting frontal slices to complete, within these slices we sample using a budget of  $m = \gamma(n^2)$  samples (adaptively or non-adaptively), where  $\gamma \in [0, 1]$  is the proportion of the total  $n^2$  entries of the slice available for sampling in the matrix completion step. Slices are then completed using the semi-definite programming formulation of the nuclear norm minimization and solved via Douglas-Rachford splitting, see [54]. In this step, convergence is declared once the primal residual, dual residual and duality gap are below  $10^{-8}$  for each of the  $s$  selected slices, or after 10,000 iterations per slice, whichever comes first. We note that the accuracy of this matrix completion step influences numerically what is achievable in terms of overall accuracy for the

completed tensor, and that the error for completing these slices is compounded in the subsequent steps (even in the absence of noise). This explains the apparent “leveling off” of the relative error even as sample complexity or signal-to-noise ratios increase. The number of fibers to sample, and which are used to estimate the remaining  $d - 2$  factor matrices is given by  $\delta r$ , where  $\delta$  is an oversampling factor of size at least one.

In Figure 4.2, we see a phase transition of our method. For a given rank, prior to a threshold, the error is dominated by the inability to accurately complete the  $s$  slices. Once sufficient samples are obtained within these slices, completion reliably succeeds and accuracy approaches the limiting numerical accuracy inherited from the initial slice completion step. We empirically see in Figure 4.2 the linear relationship of the sample complexity on the rank of the tensor - it takes approximately twice as many samples to reach this phase transition in terms of relative error when we compare the rank 5 case with the rank 10 case. In this figure,  $d = 3$ ,  $n = 200$ ,  $\alpha = 2$ ,  $s = 2$ ,  $\delta = 4$ , the parameter  $\gamma$  is varied within in the interval  $(0, 1)$  and is driving the overall sample complexity.

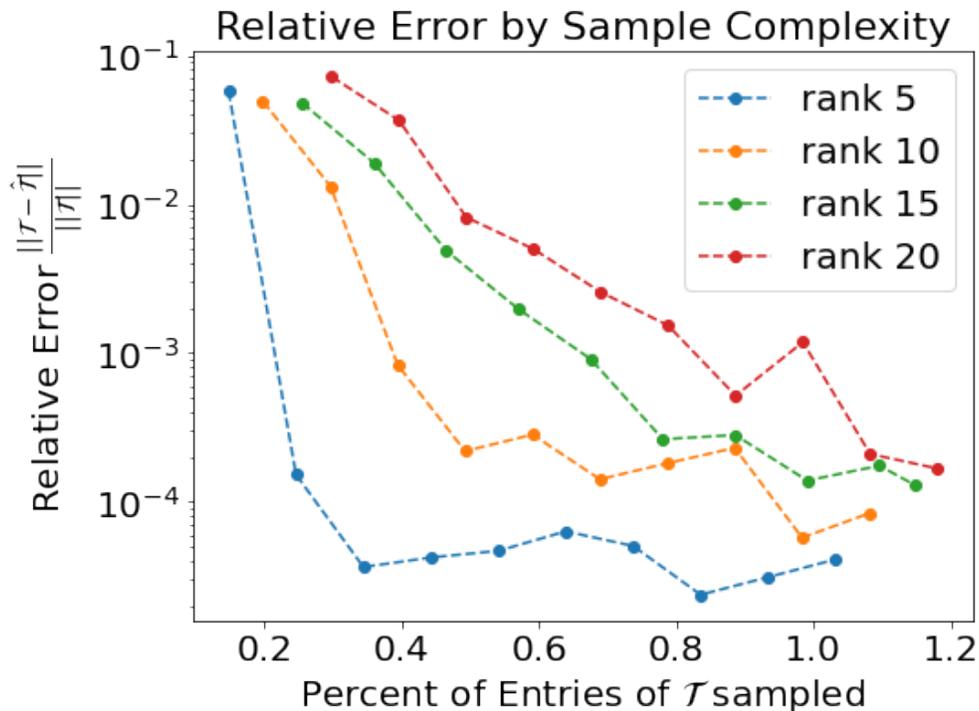


Figure 4.2 Median relative error (log-scaled) of completed tensors of varying rank as sample complexity increases without noise.

In Figure 4.3 we repeat a similar experiment, but for  $d = 4$ ,  $n = 100$ ,  $\alpha = 2$  and for three different ranks  $r = 5, 10, 15$ . These tensors are 12.5 times larger in terms of total number of entries than in the three mode case studied in Figure 4.2. Our method is able to reach relative errors at or lower than 1% using less than a tenth of one percent of the total entries. In plot (a) of Figure 4.3, we show the median relative error for Tensor Deli and the relative error after ten iteration of masked-ALS is performed on the result, i.e. the second set of experiments depend on the first in that ALS is initialized using the Tensor Deli estimates of the CP factorization, and using the same revealed entries, see [61]. As is well documented now, ALS can be quite sensitive to initialization and prone to “swamps” where accuracy stagnates over a large number of iterations. This figure shows however that Tensor Deli can provide high quality initialization, and with only a small number of iterations we can improve the accuracy of the estimate of the completed tensor by an order of magnitude without even needing to reveal more entries. Empirically we have observed ALS alone is unable to achieve comparable relative errors at this sample complexity for any observed amount of iterations, see Figure 4.5.

In plot (b) of Figure 4.3 we show the errors for adaptive vs non-adaptive strategy for each rank. In the non-adaptive case, the  $s$ -slices are sampled more heavily, but in a uniform fashion. Furthermore, no  $QR$ -factorization is performed on the first two estimated factor matrices, and instead entries are sampled uniformly throughout the rest of the tensor for the purposes of estimating the remaining factors. Not surprisingly, in the case of random low-rank tensors, the density of the sampling in the slices is driving the accuracy, and adapting the sampling pattern appears to have little effect - the slices themselves are already highly incoherent, so there is little to gain by adapting. As we discuss in the next section, on real world data with more structure, adaptive patterns do have a more measurable effect.

In Figure 4.4, we add mean-zero i.i.d. Gaussian noise to each entry in our tensor and perform the same completion strategy on the noisy tensor as described earlier in this section,  $d = 3$ ,  $n = 200$ ,  $\alpha = 2$ ,  $s = 2$ ,  $\delta = 4$ . For each trial, the noise tensor  $\mathcal{N}$  is scaled to the appropriate signal-to-noise ratio, i.e.  $\text{SNR} = 10 \log_{10} \frac{\|\mathcal{T}\|}{\|\mathcal{N}\|}$ . The sample complexity proportions are fixed with respect to slice

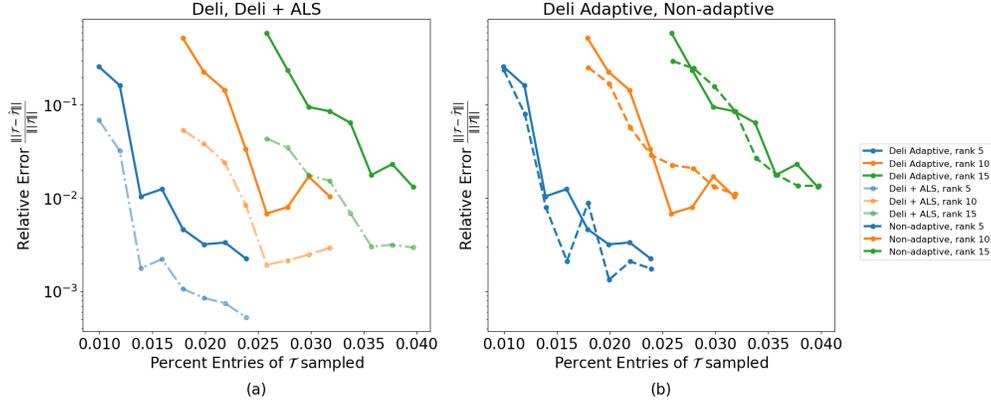


Figure 4.3 Median relative error (log-scaled) of completed four mode tensors of varying rank as sample complexity increases without noise. Each value is the median of ten trials,  $s = 2$ ,  $\gamma \in [0.1, 0.8]$ ,  $\delta = 8$ . In plot (a) we compare the relative errors of Tensor Deli before and after ten iterations of masked-ALS. In plot (b) we show error for both adaptive and non-adaptive sampling schemes.

completion versus fibers sampled to estimate  $C$  but do scale linearly according to rank in order to facilitate comparison. In all cases the total number of sampled entries is between 0.27% and 1.10% of the total number of entries, depending on rank.

As alluded to earlier, we observed during these experiments that using masked alternating least squares alone to complete our synthetic tensors (using either the Tensor Sandwich sample pattern, or an equivalent proportion of uniformly drawn samples) achieved at best relative errors of about 6%. However, the estimate of the tensor as output by Algorithm 4.2, and its sample pattern can be used as an initialization to an iterative scheme like alternating least squares to further improve the accuracy of the completed tensor. In Figure 4.5, we show three sets of related experiments: the relative error at various ranks achieved by alternating least squares alone (AS), Tensor Sandwich alone (TS), or alternating least squares initialized by Tensor Sandwich (TS + ALS). In each trial, 100 iterations of alternating least squares are used, weights for components are set to decay according to (4.3), and each of the three variations for completing the tensor are used on the same data. We notice that at this level of sample complexity, for either a sample mask chosen uniformly at random or according to the adaptive scheme described in this work, ALS alone is limited. Combined with Tensor Sandwich, however, we can see roughly an order of magnitude improvement in relative error by performing a hundred iterations of ALS on the TS estimate. This shows Tensor Sandwich can

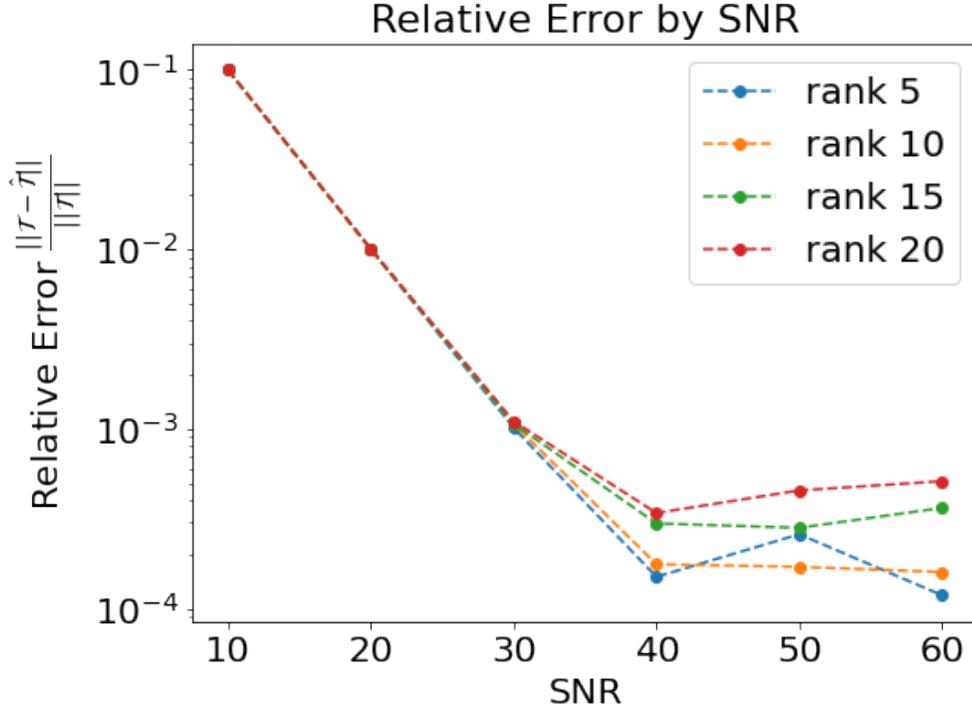


Figure 4.4 Median relative error (log-scaled) of completed tensors of varying rank for different levels of white Gaussian noise. Sample complexity is less than 1.1% for all trials, scaled linearly by rank.

be useful as an initialization strategy for other completion methods, either to save on run time by decreasing the total number of iterations needed, or to improve the accuracy of the final estimate. Here  $d = 3$ ,  $n = 200$ ,  $\alpha = 2$ ,  $s = 2$ ,  $\delta = 4$ . In all cases the total number of sampled entries is between 0.27% and 1.10% of the total number of entries, with linear scaling depending on rank to facilitate comparison.

In next pair of experiments, we compare one of the existing (non-adaptive) completion methods discussed in Subsection 4.1.2 to Tensor Sandwich. We use the same overall sample budget for Tensor Sandwich but instead use the Algorithm in [51] which we refer to as Tensor Complete (TC). The range of total values sampled is the same as in Figure 4.2. Implementation details make direct comparisons difficult. However, the empirical findings summarized in Figure 4.6 suggest that Tensor Sandwich can produce better estimates for high rank tensors than Tensor Complete can when sample budgets are limited.

In Figure 4.7 we show a comparison of one of the few adaptive methods we identified in the

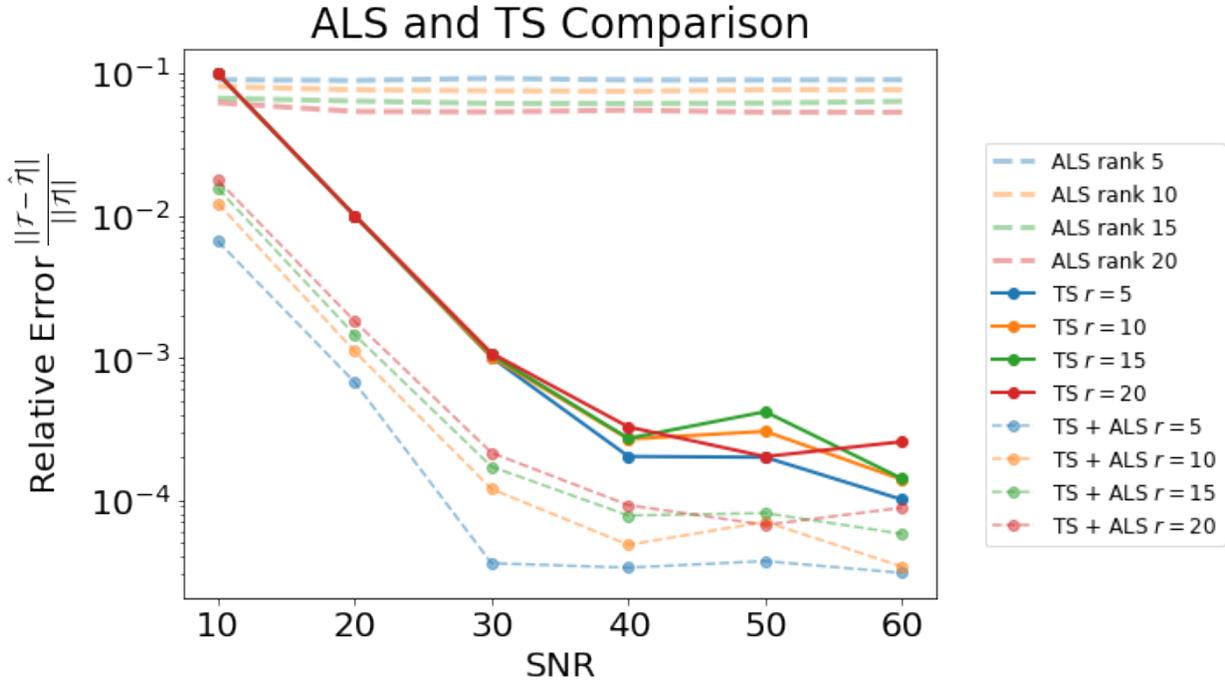


Figure 4.5 Median relative error (log-scaled) of completed tensors of varying rank for different levels of white Gaussian noise. Sample complexity is less than 1.1% for all trials, scaled linearly by rank. Tensors are completed using 100 iterations of alternating least squares (ALS), or Tensor Sandwich (TS), or 100 iterations of ALS initialized by Tensor Sandwich (TS+ALS).

area, proposed in [39] (denoted KS in the figure) and adaptive Tensor Sandwich for  $d = 3$ ,  $n = 40$ ,  $\alpha = 1$  with ten iterations of post-ALS and for three different ranks  $r = 4, 6, 8$ . In plot (a) we have along the horizontal axis the two respective parameters for each of the methods that drives their overall sample complexity. For KS, the sample complexity is largely driven by the proportion of entries the method is able to sample in a given fiber to test whether that fiber should be added to the basis. For Tensor Deli, this is  $\gamma$ , the proportion of entries in a single face the method is able to sample when completing that slice. In the figure we note Tensor Deli exhibits a phase transition for each rank where after this threshold of a certain size for  $\gamma$  the method achieves much better relative error. That this threshold changes in an approximately linear fashion with respect to rank supports out theoretical conclusions. Plot (b) in Figure 4.7 changes the horizontal axis to total percent of the tensor sampled, and we can now plainly observe that, for a smaller overall sample budget by a factor of least four, Tensor Deli is able to complete the data more accurately than KS. It is also the case that Tensor Deli outputs an estimate that is truly a CP-tensor of the chosen rank truncation,

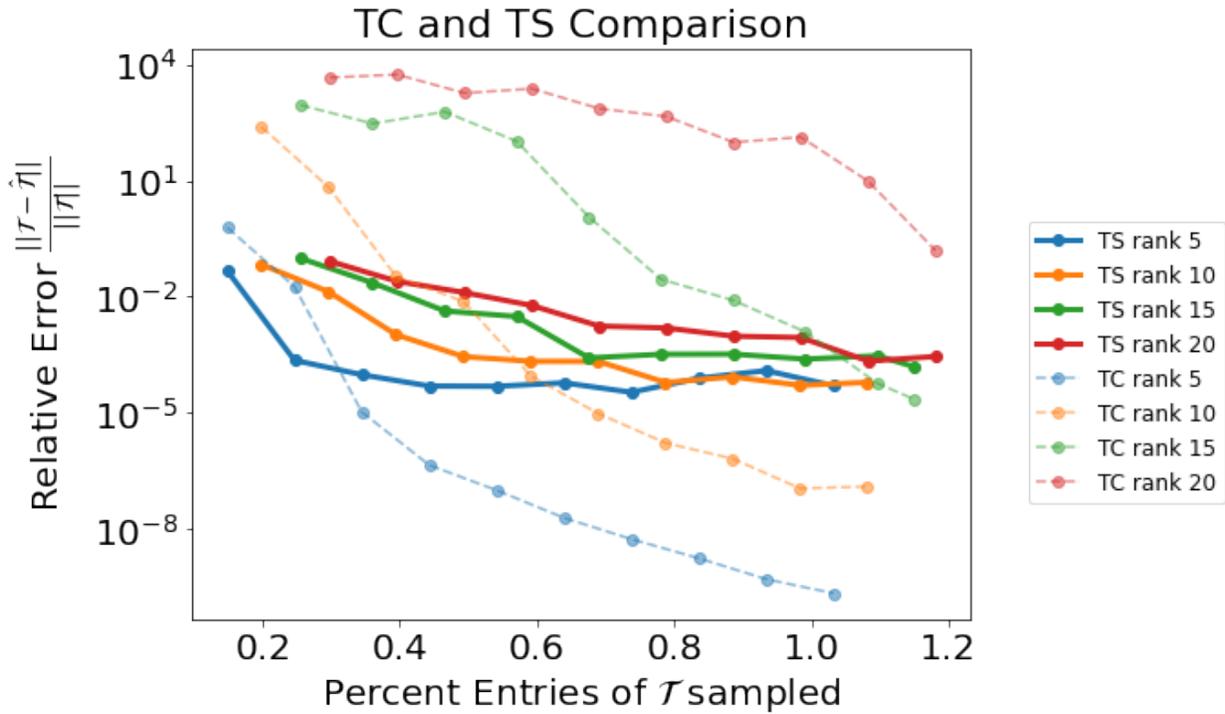


Figure 4.6 Median relative error (log-scaled) of completed tensors of varying rank for different total number of entries revealed. Tensors are completed using Tensor Sandwich (TS), or Tensor Complete as described in [51] without noise.

whereas KS in general will not output a tensor that has a CP-decomposition of the selected rank, but instead a more complicated recursive decomposition. CP fitting can be conducted on the KS estimate, for example using iterations of ALS, but this in general will reduce the accuracy of the estimate.

### KS and Tensor Deli Comparison

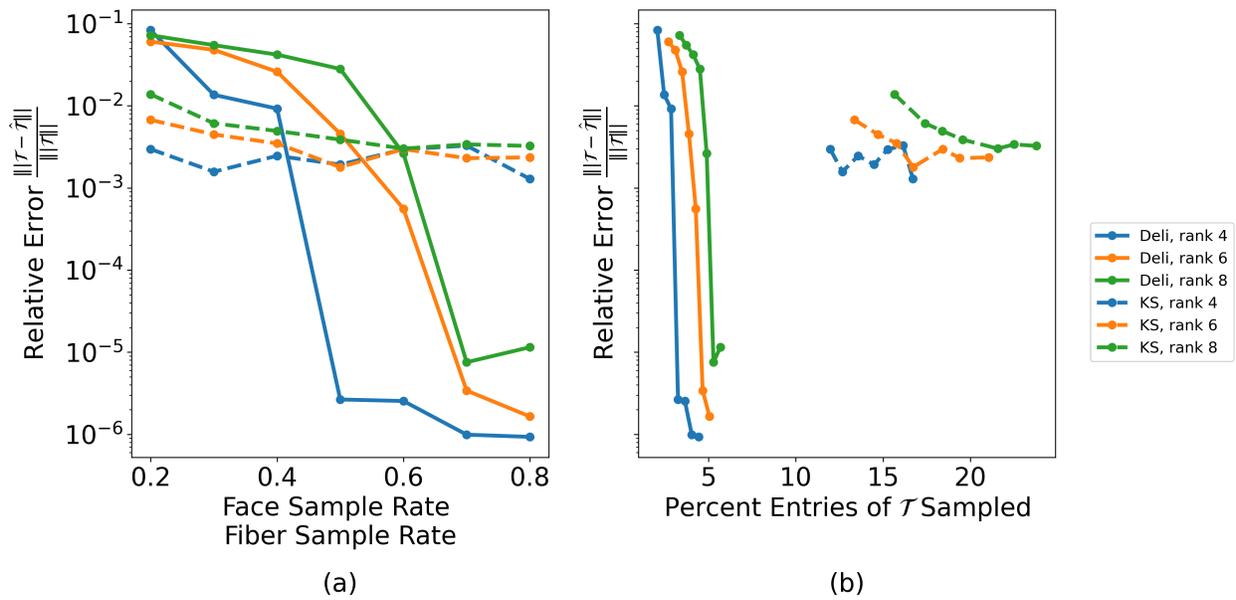


Figure 4.7 Median relative error (log-scaled) of completed three mode tensors of varying rank without noise for our proposed Tensor Deli with adaptive sampling method, and the adaptive algorithm proposed in [39]. Each value is the median of ten trials, for Tensor Deli  $s = 2, \gamma \in [0.2, 0.8], \delta = 8$ , for KS the proportion of entries that can be sampled for a fiber range from  $[0, 2, 0.8]$ . In plot (a) we show the relative error by the face or fiber sampling parameter for each of the methods.

### 4.3.2 Applications

We also apply Tensor Sandwich on data-sets in following application areas: chemo-metrics, and hyper-spectral imaging. The chemo-metrics data-set is as used in [7]. It is fluorescence measured from known analytes and the data is intended for calibration purposes. There are a total of 405 fluorophores of six different types. For each sample an Excitation Emission Matrix (EEM) was measured - i.e. fluorescence intensity was collected for different set levels of excitation wavelength and emission wavelength. The EEM of an unknown sample can be used for its identification and studying other properties of interest for a given analyte. In this data-set there are a 136 emission wavelengths and 19 excitation wavelengths. In the original tensor, there are five replicates per sample, however in our experiment we discarded all but the first set of replicates and so formed then a three mode tensor of size  $405 \times 136 \times 19$ . We compare the completed tensor for five different choices of rank,  $r = 11, 13, 15, 17, 19$ . We set  $s = 4$ , and thus complete four frontal slices. We fix frontal slices where the third mode is at indices 5, 9, 13, 17 across all experiments in order to facilitate better comparison. In Figure 4.8, we show the recovered tensor for a representative frontal slice at each of the different ranks. Below this in the same figure, we have fixed a representative lateral slice, which corresponds to a completed EEM for a particular sample of an analyte at the different ranks. In all cases, the total number of entries revealed is between 10 and 11% of the tensor's entries, we performed adaptive sampling and ten iterations of ALS on the resulting CP estimate, sampling parameters  $\gamma$  and  $\delta$  are 0.5 and 10 respectively.

Furthermore, in Figure 4.9 we show evidence that indeed, in practical applications unlike the simulated data, there is quite clearly a benefit to sample adaptively in terms of the trade off between accuracy and sample complexity. In this Figure, for a fixed rank of 15 and a fixed sample complexity using four slices and the sampling parameters of 0.5 and 10 for  $\gamma$  and  $\delta$ , using the same lateral slice as before in Figure 4.8, we show the completed EEM and corresponding relative error for this slice using the adaptive Tensor Deli (Algorithm 4.3), the adaptive Tensor Deli with ten iterations of ALS, and non-adaptive Tensor Deli. In this case, the additional ALS iterations show only modest improvement to the overall estimates, however the adaptive scheme's accuracy has a much larger

effect. This is likely do to the fact the data's coherence and other parameters related to the problem are far from ideal, especially when compared to the simulated case from earlier.

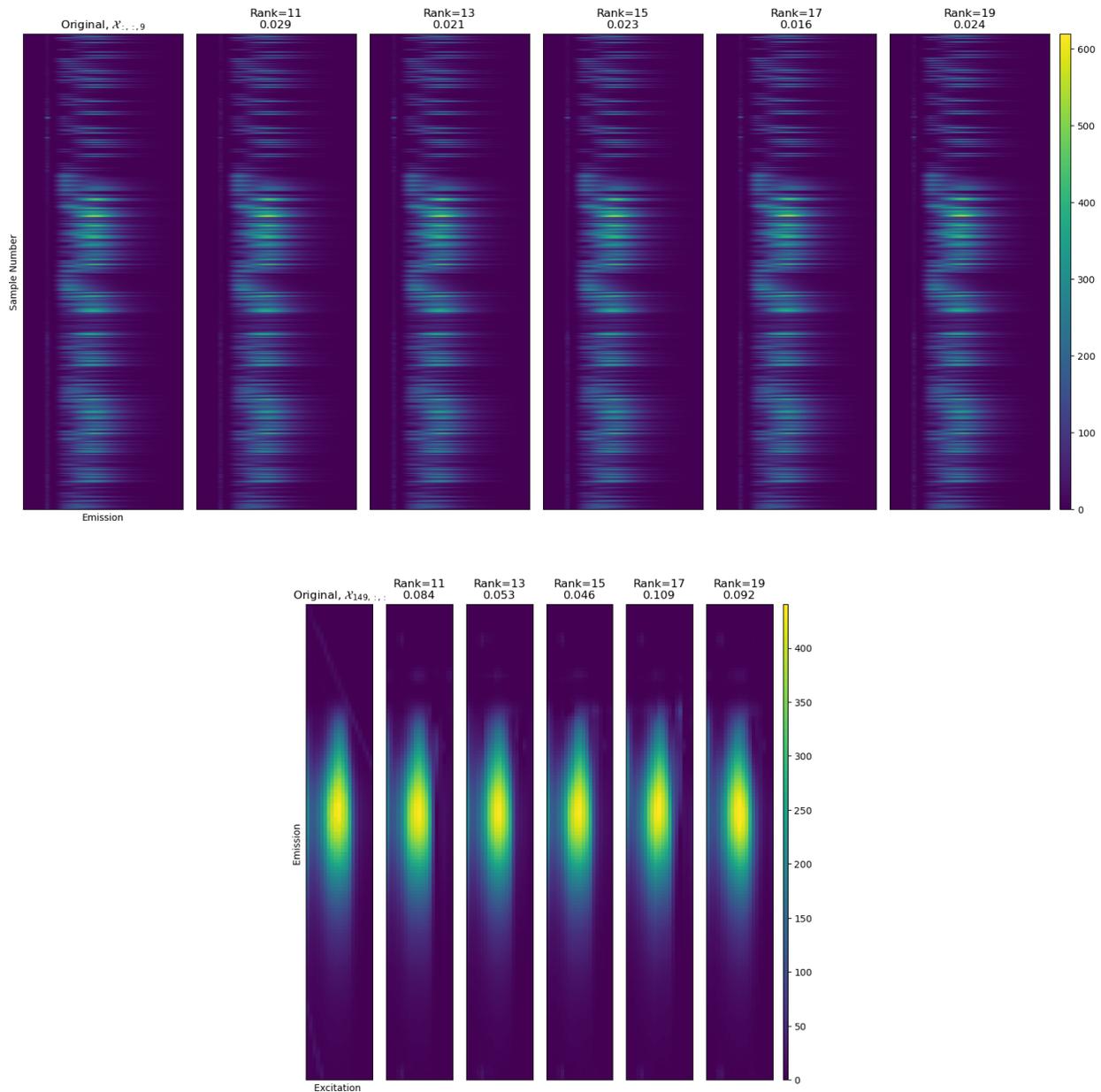


Figure 4.8 Slices of a completed tensor for fluorescence data. The top row shows the original frontal slice  $\mathcal{X}_{:, :, 9}$  and the same slice completed at ranks 11, 13, 15, 17, 19 with the corresponding relative error. The bottom row shows a lateral slice  $\mathcal{X}_{149, :, :}$ , which corresponds then to the EEM for the single analyte number 149 in the dataset.

The next application is hyperspectral imaging data. The data is as found in [6]. The hyperspectral sensor data was acquired in June 1992 and consists of aerial images of an approximately two mile

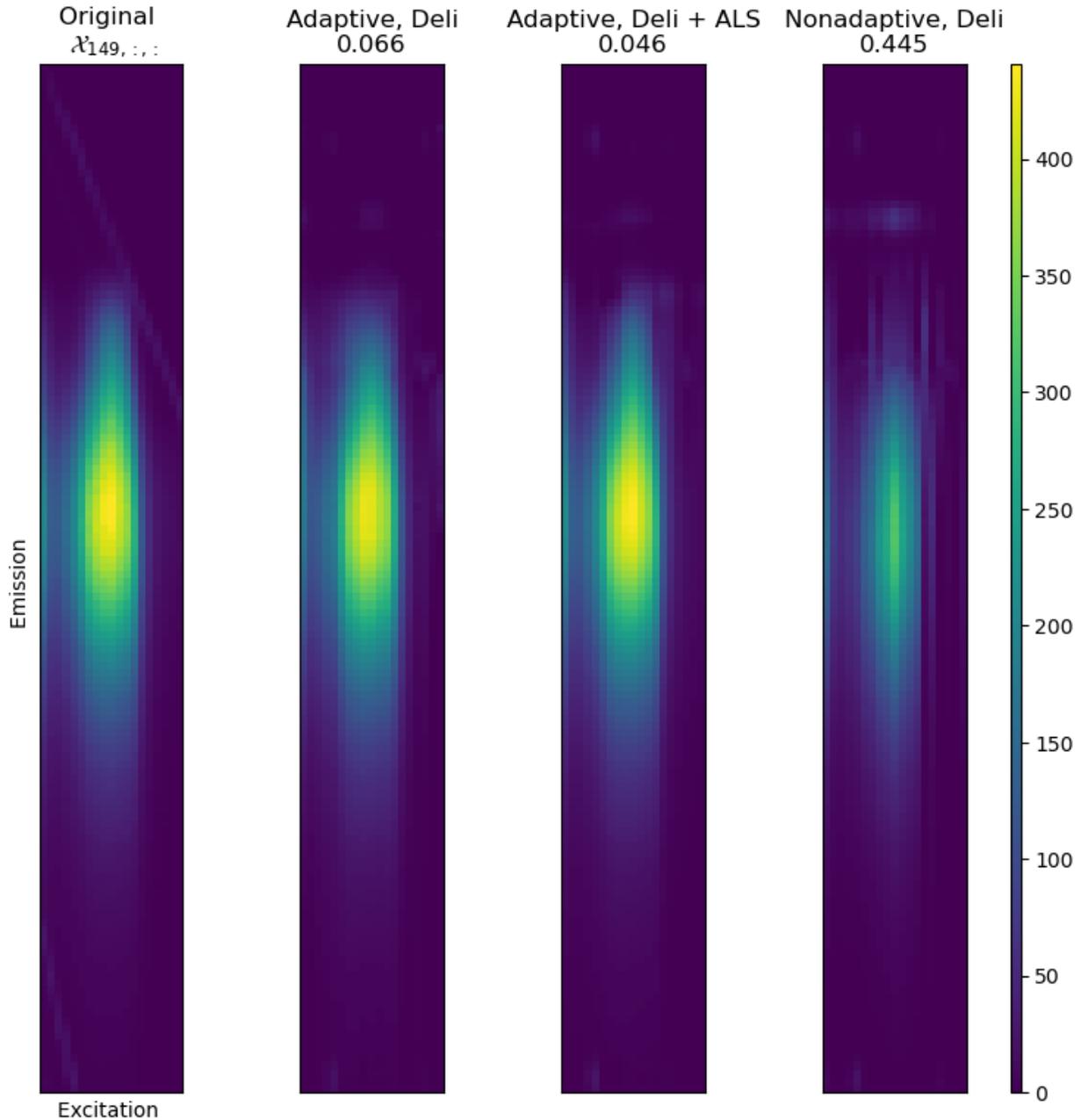


Figure 4.9 Lateral slice corresponding to the EEM for analyte number 149 in the fluorescence data is compared with the corresponding slice completed from adaptive, adaptive with extra iterations of ALS, and non-adaptive Tensor Deli. Relative error for the slice is listed for each method. Rank is fixed at 15 for each, sample complexity is fixed at about 10% of entries using sampling parameters of 0.5 and 10 for  $\gamma$  and  $\delta$  respectively.

by to mile Purdue University Agronomy farm and was originally intended for soils research. The data consists of 200 images at different wave lengths that are 145 by 145 pixels each. We thus form a three mode,  $145 \times 145 \times 200$  data tensor. We complete the tensor using  $s = 9$ ,

where we have fixed these frontal slices at indices 20, 40, 60, 80, 100, 120, 140, 160, 180 across all experiments to facilitate comparison. Shown in Figure 4.10, we have completed the tensor for ranks  $r = 20, 30, 40, 50, 60, 80$  and displayed a fixed, representative frontal slice at index 48. The first row consists of data completed using Tensor Deli with adaptive sampling, with parameters  $\gamma = 0.7$  and  $\delta = 10$  for a total sample budget that is about 4.5% of the total entries. The second row then performs ten iterations of masked-ALS using the initialization of the top row and the same revealed entries. The last row consists of ten iterations masked-ALS, which is initialized using the SVD of the flattenings, e.g. see [61].

Empirically, we observe the hyperspectral image dataset is imperfectly approximated by a low-rank CP decomposition to begin with. In Figure 4.10, we see Tensor Deli alone performs in terms of global relative error certainly no better than masked-ALS alone on this data. However, it provides a superior initialization to ALS, as we can see in the second row. Moreover, we observe there is a qualitative difference in the completed slices and their errors. In the ALS alone case, the error appears achieved by a sort of local averaging of intensity of the pixels whereas Tensor Deli does capture contrasts and local features more distinctly, and the errors are largest on rows and columns it “misses” or imperfectly completes. This can be seen by looking at a heat map of the error slice by slice. Looking at the middle row, we can see there is a distinct advantage of combining these two types of methods to achieve the best sort of completion for this dataset.

#### 4.4 Discussion

We have presented an adaptive sampling approach for low CP-rank tensor completion which completes a CP-rank  $r$  tensor of size  $n \times n \times n$  using  $\mathcal{O}(nr \log^2 r)$  samples with high probability. Our method significantly improves on the tensor completion result in [39] while only making a mild additional assumption on the third factor matrix. We also provided numerical experiments to demonstrate that a version of our tensor completion method is robust to noise empirically, works as a high quality initialization to ALS, and can run efficiently on tensors of more than three modes, in the regime of hundreds of millions of entries.

A non-adaptive version of our tensor completion algorithm is also possible, as suggested in

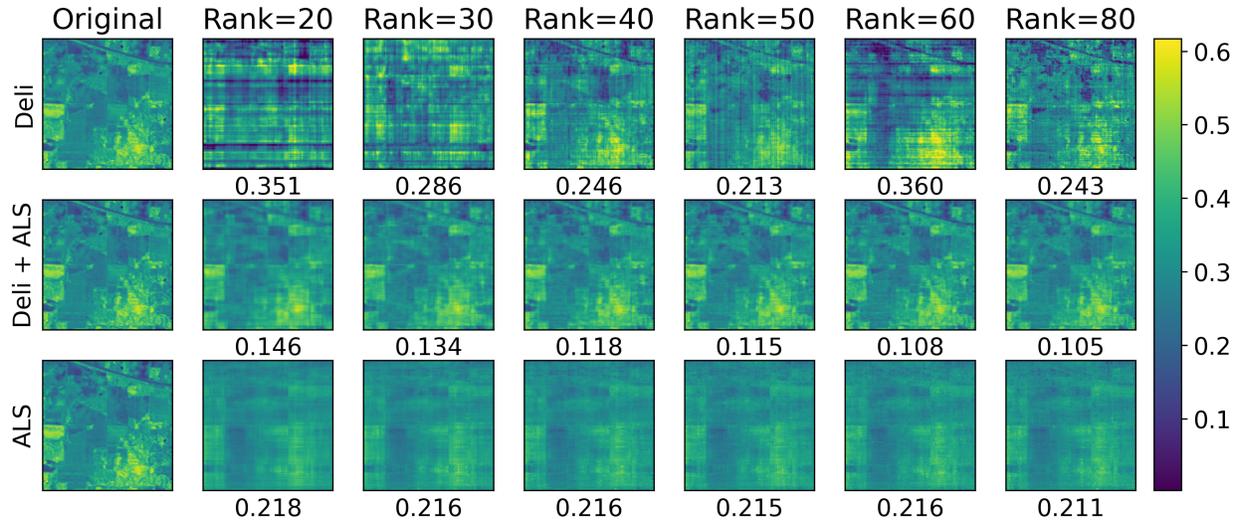


Figure 4.10 Slices of a completed tensor for hyperspectral data. The first column shows the original frontal slice  $\mathcal{X}_{:, :, 48}$  and the same slice completed at ranks 20, 30, 40, 50, 60, 80 with the corresponding relative error below for three different methods. The top row is adaptive Tensor Deli with sampling parameters  $\gamma = 0.7$ ,  $\delta = 10$ , the middle row shows the slice after ten iterations of ALS using the Deli initialization and the last row shows SVD initialized masked-ALS.

Section 4.5. Second, it is possible to extend our tensor completion method to higher-order tensors, as we outline. We show the result for the case where factor matrices do not have zero values, as this illuminates very clearly what system of equations must be consistent to ensure exact recovery of the factor matrix, however it is possible to remove this assumption, and instead deal with the case of zeroes in the factor matrices using (a few) more samples.

#### 4.5 Extending to Higher Order Tensors

The purpose of this section is to include details on how the proposed method can be extended to tensors of more than three modes. Within this proof we also see how the method can be made non-adaptive; by increasing the sample complexity to meet the necessary bounds for the matrix completion and censored least square solves with high probability.

---

**Algorithm 4.3** Adaptive Tensor Deli for Order- $d$  Tensors with No Zeros in  $A^{(3)}, \dots, A^{(d)}$

---

**input** :  $S \subseteq [n_3]$  slices to complete

**output** :  $\hat{T}$ , completed tensor

# Slice Complete Phase

**for**  $k \in S$  **do**

    Use, e.g., [10], or algorithm 1 in [40] to complete  $\mathcal{T}_{::,k,1,\dots,1}$ . Algorithm 1 in [40] uses at most  $C'_1 \mu_0 \max(n_1, n_2) r \log^2(r^2/\delta)$  adaptively chosen samples.

**end**

# Jennrich Complete Phase

Generate random vectors  $\mathbf{u}, \mathbf{v} \in \mathbb{S}^{s-1}$ ;

$T_u \leftarrow \sum_{k \in S} u_k T_{::,k,1,\dots,1}$

$T_v \leftarrow \sum_{k \in S} v_k T_{::,k,1,\dots,1}$

Compute eigen-decomposition  $A^{(1)} \mathbf{\Lambda} (A^{(1)})^{-1} = T_u (T_u)^\dagger$

$A^{(2)} \leftarrow (A^{(1)})^{-1} T_u$

# Censored Least Squares Phase to learn modes 3 through  $d$

**for**  $k = 3, 4, \dots, d$  **do**

$\Omega_{k-1} \leftarrow \text{fiber sampler}(A^{(1)}, A^{(2)}, r, \gamma)$

$(\tilde{A}^{(3)})^T \leftarrow \text{censored least squares}((A^{(1)} \odot A^{(2)}), \mathcal{T}_{i_1, i_2, 1, \dots, 1, i_k, 1, \dots, 1} \text{ s.t. } (i_1, i_2, 1, \dots, 1, i_k, 1, \dots, 1) \in \Omega_{k-1})$

$\Omega_{k-1}$

$A^{(k)} \leftarrow \tilde{A}^{(k)}$  with columns normalized to unit length

**end**

$\alpha \leftarrow \text{censored least squares}((A^{(1)} \odot A^{(2)} \dots \odot A^{(d)}), \text{vec}(\mathcal{T}_i) \text{ s.t. } \mathbf{i} \in \bigcup_{k=1}^{d-1} \Omega_{k-1})$

$\hat{\mathcal{T}} \leftarrow [[\mathbf{a}, A^{(1)}, \dots, A^{(d)}]]$

---

**Theorem 4.2.** *Suppose that the factor matrices satisfy the following assumptions for CP rank- $r$  tensor  $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times \dots \times n_d}$*

- $\text{rank}(A^{(1)}) = \text{rank}(A^{(2)}) = r$
- $\mu(A^{(1)}) \leq \mu_0$
- every  $s \times r$  submatrix of  $A^{(3)}$  has Kruskal rank  $\geq 2$
- for  $k = 3, \dots, d$ ,  $A^{(k)}$  has no entries which are 0.

*Then, with probability at least  $1 - s\delta$ , Algorithm 3 completes  $\mathcal{T}$  and uses at most*

$$s \left( C \mu_0 n r \log^2(r^2/\delta) \right) + r \sum_{k=3}^d n_k$$

*samples,  $n = \max(n_1, n_2)$*

*Proof.* We can prove this result by completing well chosen 3-mode sub-tensors. Consider first the 3-mode sub-tensor  $\mathcal{T}_{i_1, i_2, i_3, \mathbf{i}}$ , for  $i_j \subseteq [n_j]$ ,  $\mathbf{i}$  the index vector of all ones of length  $d - 3$ ,  $[1]^{d-3}$ . By our assumptions and Theorem 4.1, using the corresponding set of entries as previously described in the proof of Theorem 4.1 only where modes beyond three are fixed at index 1, we can define sets  $\Omega_1 \subset [n_1] \times [n_2] \times [s_k]$ , the set of indices needed to complete the  $s$  selected (frontal) slices for matrix completion, where  $s_k \in S$ ,  $S \subset [n_3]$  and used to find factor matrices  $A^{(1)}$  and  $A^{(2)}$ , and  $\Omega_2 \subset [n_1] \times [n_2] \times [n_3] \times [1] \times \dots \times [1]$  where  $(i_1, i_2, i_3, 1, \dots, 1)$  s.t.  $(i_1, i_2)$  are selected from the  $QR$  decomposition of  $(A^{(1)} \odot A^{(2)})^T$ ,  $i_3$  ranges over all  $[n_3]$  are the set of fibers needed we can complete the 3-mode sub-tensor and obtain factor matrices  $\tilde{A}^{(3)}$ . That is, using the same procedure as the three mode case, we have solved the system of equations for  $\{\tilde{A}_{i_3, \ell}^{(3)}\}_{\ell=1}^r$ ,

$$\begin{aligned} \sum_{\ell=1}^r (A_{i_1, \ell}^{(1)} A_{i_2, \ell}^{(2)} A_{i_3, \ell}^{(3)}) (A_{1, \ell}^{(4)} \cdots A_{1, \ell}^{(d)}) &= \sum_{\ell=1}^r A_{i_1, \ell}^{(1)} A_{i_2, \ell}^{(2)} \tilde{A}_{i_3, \ell}^{(3)} \\ &= \mathcal{T}_{i_1, i_2, i_3, 1, 1, \dots, 1} \end{aligned}$$

Where  $\tilde{A}_{i_3, \ell}^{(3)} = A_{i_3, \ell}^{(3)} A_{1, \ell}^{(4)} \cdots A_{1, \ell}^{(d)}$ , i.e. the scaled version of  $A^{(3)}$ , importantly, because the factor matrices are assumed to be entirely non-zero,  $A_{1, \ell}^{(4)}$  and so on will all be non-zero, and so we are guaranteed a non-zero and thus informative value for  $A^{(3)}$  in all the  $r$  selected equations.

We remark, here suggests immediately how we can relax the last assumption concerning the factor matrices for modes 4 and higher. Should the factor matrices in fact have zeroes, we can with high probability get  $r$  informative equations by sampling more.

We can now permute the indices, and solve a similar looking system for the similarly defined  $\tilde{A}^{(4)}$ ,  $(i_1, i_2)$  same as before,  $i_4 \in [n_4]$  and so indices  $(i_1, i_2, 1, i_4, 1, \dots, 1) \in \Omega_3$  correspond to  $r$  mode-4 fibers of the tensor. We have the system

$$\begin{aligned} \sum_{\ell=1}^r (A_{i_1, \ell}^{(1)} A_{i_2, \ell}^{(2)} A_{i_4, \ell}^{(4)}) (A_{1, \ell}^{(3)} \cdot A_{1, \ell}^{(5)} \cdots A_{1, \ell}^{(d)}) &= \sum_{\ell=1}^r A_{i_1, \ell}^{(1)} A_{i_2, \ell}^{(2)} \tilde{A}_{i_4, \ell}^{(4)} \\ &= \mathcal{T}_{i_1, i_2, 1, i_4, 1, \dots, 1} \end{aligned}$$

We repeat this procedure for all remaining factor matrices. A final solve to rectify the scaling indeterminacy and obtain the weights of the rank-1 components as the vector  $\alpha \in \mathbb{R}^r$ , i.e. solve the censored least squares

$$(A^{(1)} \odot A^{(2)} \odot \tilde{A}^{(1)} \odot \cdots \odot \tilde{A}^{(d)}) \alpha = \text{vec}(\mathcal{T})_{\Omega}$$

$\Omega = \bigcup_{k=1}^{d-1} \Omega_k$ . To show then the sample complexity bound, we need to bound  $|\Omega|$ . We have  $r$  fibers in  $d - 2$  modes indexed by  $(i_1, i_2, 1, \dots, i_k, \dots, 1)$ , for each  $k = 3, 4, \dots, d$ ,  $i_k \in [n_k]$  and these indices are in  $\Omega_{k-1}$ . I.e.  $(d - 2)r$  fibers of length  $n_i$  for a total of  $r \sum_{i=3}^d n_i$  revealed indices from fiber sampling once we have union bounded over all the sets  $\Omega_k$ . This is in addition (but again potentially intersecting) with the revealed entries used to completed the  $s$  slices, the elements of set  $\Omega_1$ . A final union bound yields the desired sample complexity and failure probability.

□

## BIBLIOGRAPHY

- [1] Thomas D. Ahle and Jakob B. T. Knudsen. Almost optimal tensor sketch, 2019.
- [2] Rosa I Arriaga and Santosh Vempala. An algorithmic theory of learning: Robust concepts and random projection. *Machine learning*, 63:161–182, 2006.
- [3] Stefan Bamberger, Felix Krahmer, and Rachel Ward. Johnson–lindenstrauss embeddings with kronecker structure. *SIAM Journal on Matrix Analysis and Applications*, 43(4):1806–1850, 2022.
- [4] Boaz Barak and Ankur Moitra. Noisy tensor completion via the sum-of-squares hierarchy. In *Conference on Learning Theory*, pages 417–445. PMLR, 2016.
- [5] Richard G Baraniuk, Mark A Davenport, Ronald A DeVore, and Michael B Wakin. A simple proof of the restricted isometry property for random matrices. *Constructive Approximation*, 2007.
- [6] Marion F. Baumgardner, Larry L. Biehl, and David A. Landgrebe. 220 band aviris hyper-spectral image data set: June 12, 1992 indian pine test site 3, Sep 2015.
- [7] Rasmus Bro, Åsmund Rinnan, and Nicolaas Faber. Standard error of prediction for multilinear pls - 2. practical implementation in fluorescence spectroscopy. *Chemometrics and Intelligent Laboratory Systems*, 75:69–76, 01 2005.
- [8] Emmanuel Candes and Benjamin Recht. Exact matrix completion via convex optimization. *Communications of the ACM*, 55(6):111–119, 2012.
- [9] Hongyang Chen, Fauzia Ahmad, Sergiy Vorobyov, and Fatih Porikli. Tensor decompositions in wireless communications and mimo radar. *IEEE Journal of Selected Topics in Signal Processing*, 15(3):438–453, 2021.
- [10] Yudong Chen, Srinadh Bhojanapalli, Sujay Sanghavi, and Rachel Ward. Completing any low-rank matrix, provably. *Journal of Machine Learning Research*, 16(94):2999–3034, 2015.
- [11] Agniva Chowdhury, Jiasen Yang, and Petros Drineas. Structural conditions for projection-cost preservation via randomized matrix multiplication. *Linear Algebra and its Applications*, 573:144–165, 2019.
- [12] Andrzej Cichocki. Tensor decompositions: new concepts in brain data analysis? *Journal of the Society of Instrument and Control Engineers*, 50(7):507–516, 2011.
- [13] Michael B Cohen, Sam Elder, Cameron Musco, Christopher Musco, and Madalina Persu. Dimensionality reduction for k-means clustering and low rank approximation. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 163–172, 2015.
- [14] Mark A Davenport and Justin Romberg. An overview of low-rank matrix recovery from incomplete observations. *IEEE Journal of Selected Topics in Signal Processing*, 10(4):608–622, 2016.

- [15] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.
- [16] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. On the best rank-1 and rank- $(r_1, r_2, \dots, r_n)$  approximation of higher-order tensors. *SIAM journal on Matrix Analysis and Applications*, 21(4):1324–1342, 2000.
- [17] Dan Feldman, Melanie Schmidt, and Christian Sohler. Turning big data into tiny data: Constant-size coresets for k-means, pca, and projective clustering. *SIAM Journal on Computing*, 49(3):601–657, 2020.
- [18] Simon Foucart and Holger Rauhut. *A Mathematical Introduction to Compressive Sensing. Applied and Numerical Harmonic Analysis*. Springer New York, New York, NY, 2013.
- [19] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, third edition, 1996.
- [20] Rachel Grotheer, Shuang Li, Anna Ma, Deanna Needell, and Jing Qin. Iterative hard thresholding for low CP-rank tensor models. *arXiv preprint arXiv:1908.08479*, 2019.
- [21] Rachel Grotheer, Anna Ma, Deanna Needell, Shuang Li, and Jing Qin. Stochastic iterative hard thresholding for low Tucker rank tensor recovery. In *Proc. Information Theory and Applications*, 2020.
- [22] Wolfgang Hackbusch. *Tensor Spaces and Numerical Tensor Calculus. [electronic resource]*. Springer Series in Computational Mathematics: 56. Springer International Publishing, 2019.
- [23] Cullen Haselby, Mark Iwen, Deanna Needell, Michael Perlmutter, and Elizaveta Rebrova. Modewise operators, the tensor restricted isometry property, and low-rank tensor recovery. *Applied and Computational Harmonic Analysis*, 66:161–192, 2023.
- [24] Cullen Haselby, Mark Iwen, Deanna Needell, Elizaveta Rebrova, and William Swartworth. Fast and low-memory compressive sensing algorithms for low tucker-rank tensor approximation from streamed measurements, 2023.
- [25] Cullen Haselby, Santhosh Karnik, and Mark Iwen. Tensor sandwich: Tensor completion for low CP-rank tensors via adaptive random sampling. In *Fourteenth International Conference on Sampling Theory and Applications*, 2023.
- [26] Stijn Hendriks and Lieven De Lathauwer. Block row kronecker-structured linear systems with a low-rank tensor solution. *Frontiers in Applied Mathematics and Statistics*, 8, 2022.
- [27] Mikael Møller Høgsgaard, Lion Kamma, Kasper Green Larsen, Jelani Nelson, and Chris Schwiiegelshohn. Sparse dimensionality reduction revisited, 2023.
- [28] Mark Iwen, Deanna Needell, Elizaveta Rebrova, and Ali Zare. Lower memory oblivious (tensor) subspace embeddings with fewer random bits: modewise methods for least squares. *SIAM Journal on Matrix Analysis and Applications*, 42(1):376–416, 2021.

- [29] Prateek Jain and Sewoong Oh. Provable tensor factorization with missing data. *Advances in Neural Information Processing Systems*, 27, 2014.
- [30] Ruhui Jin, Tamara G. Kolda, and Rachel Ward. Faster johnson-lindenstrauss transforms via kronecker products. *CoRR*, abs/1909.04801, 2019.
- [31] Ten Berge JM. The  $k$ -rank of a khatri-rao product. *Unpublished Note, Heijmans Institute of Psychological Research, University of Groningen, The Netherlands.*, 2000.
- [32] William B Johnson. Extensions of lipschitz mappings into a hilbert space. *Contemp. Math.*, 26:189–206, 1984.
- [33] Michael Kapralov, Rasmus Pagh, Ameya Velingker, David P. Woodruff, and Amir Zandieh. Oblivious sketching of high-degree polynomial kernels. *CoRR*, abs/1909.01410, 2019.
- [34] Bohdan Kivva and Aaron Potechin. Exact nuclear norm, completion and decomposition for random overcomplete tensors via degree-4 sos. *arXiv preprint arXiv:2011.09416*, 2020.
- [35] Andrew V. Knyazev and Merico E. Argentati. Principal angles between subspaces in an  $a$ -based scalar product: Algorithms and perturbation estimates. *SIAM Journal on Scientific Computing*, 23(6):2008–2040, 2002.
- [36] Tamara G. Kolda. Will the real jennrich’s algorithm please stand up? <https://www.mathsci.ai/post/jennrich/>. Accessed: 2023-05-23.
- [37] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.
- [38] Felix Kraher and Rachel Ward. New and improved johnson–lindenstrauss embeddings via the restricted isometry property. *SIAM Journal on Mathematical Analysis*, 43(3):1269–1281, 2011.
- [39] Akshay Krishnamurthy and Aarti Singh. Low-rank matrix and tensor completion via adaptive sampling. *Advances in neural information processing systems*, 26, 2013.
- [40] Akshay Krishnamurthy and Aarti Singh. On the power of adaptivity in matrix completion and approximation, 2014.
- [41] Rafał Latała. Estimation of moments of sums of independent real random variables. *The Annals of Probability*, 25(3):1502 – 1513, 1997.
- [42] Denis Le Bihan, Jean-François Mangin, Cyril Poupon, Chris A Clark, Sabina Pappata, Nicolas Molko, and Hughes Chabriat. Diffusion tensor imaging: concepts and applications. *Journal of Magnetic Resonance Imaging: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 13(4):534–546, 2001.
- [43] Michel Ledoux and Michel Talagrand. *Probability in Banach Spaces : Isoperimetry and Processes*. Applied and Numerical Harmonic Analysis. Springer New York, 2002.

- [44] S. E. Leurgans, R. T. Ross, and R. B. Abel. A decomposition for three-way arrays. *SIAM Journal on Matrix Analysis and Applications*, 14(4):1064–1083, 1993.
- [45] Allen Liu and Ankur Moitra. Tensor completion made practical. *Advances in Neural Information Processing Systems*, 33:18905–18916, 2020.
- [46] Avner Magen and Anastasios Zouzias. Low rank matrix-valued chernoff bounds and approximate matrix multiplication. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 1422–1436. SIAM, 2011.
- [47] Osman Asif Malik and Stephen Becker. Low-rank tucker decomposition of large tensors using tensorsketch. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.
- [48] Per-Gunnar Martinsson, Vladimir Rokhlin, and Mark Tygert. A randomized algorithm for the decomposition of matrices. *Applied and Computational Harmonic Analysis*, 30(1):47–68, 2011.
- [49] Jiří Matoušek. On variants of the johnson–lindenstrauss lemma. *Random Structures & Algorithms*, 33(2):142–156, 2008.
- [50] Rachel Minster, Arvind K. Saibaba, and Misha E. Kilmer. Randomized algorithms for low-rank tensor decompositions in the tucker format. 2019.
- [51] Ankur Moitra. *Algorithmic aspects of machine learning*. Cambridge University Press, 2018.
- [52] Andrea Montanari and Nike Sun. Spectral algorithms for tensor completion. *Communications on Pure and Applied Mathematics*, 71(11):2381–2425, 2018.
- [53] Cameron Musco and Christopher Musco. Projection-cost-preserving sketches: Proof strategies and constructions. *CoRR*, abs/2004.08434, 2020.
- [54] Brendan O’Donoghue, Eric Chu, Neal Parikh, and Stephen Boyd. Conic optimization via operator splitting and homogeneous self-dual embedding. *Journal of Optimization Theory and Applications*, 169(3):1042–1068, June 2016.
- [55] Aaron Potechin and David Steurer. Exact tensor completion with sum-of-squares. In *Conference on Learning Theory*, pages 1619–1673. PMLR, 2017.
- [56] Holger Rauhut, Reinhold Schneider, and Željka Stojanac. Low rank tensor recovery via iterative hard thresholding. *Linear Algebra and its Applications*, 523:220–262, 2017.
- [57] Tamas Sarlos. Improved approximation algorithms for large matrices via random projections. In *2006 47th annual IEEE symposium on foundations of computer science (FOCS’06)*, pages 143–152. IEEE, 2006.
- [58] Mikael Sørensen and Lieven De Lathauwer. Fiber sampling approach to canonical polyadic decomposition and application to tensor completion. *SIAM Journal on Matrix Analysis and Applications*, 40(3):888–917, 2019.

- [59] Jimeng Sun, Dacheng Tao, Spiros Papadimitriou, Philip S Yu, and Christos Faloutsos. Incremental tensor analysis: Theory and applications. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 2(3):1–37, 2008.
- [60] Yiming Sun, Yang Guo, Charlene Luo, Joel Tropp, and Madeleine Udell. Low-rank tucker approximation of a tensor from streaming data. *SIAM Journal on Mathematics of Data Science*, 2(4):1123–1150, 2020.
- [61] Giorgio Tomasi and Rasmus Bro. Parafac and missing values. *Chemometrics and Intelligent Laboratory Systems*, 75(2):163–180, 2005.
- [62] Nick Vannieuwenhoven, Raf Vandebril, and Karl Meerbergen. A new truncation strategy for the higher-order singular value decomposition. *SIAM Journal on Scientific Computing*, 34(2):A1027–A1052, 2012.
- [63] Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2018.
- [64] Franco Woolfe, Edo Liberty, Vladimir Rokhlin, and Mark Tygert. A fast randomized algorithm for the approximation of matrices. *Applied and Computational Harmonic Analysis*, 25(3):335–366, 2008.
- [65] Ming Yuan and Cun-Hui Zhang. On tensor completion via nuclear norm minimization. *Foundations of Computational Mathematics*, 16(4):1031–1068, 2016.
- [66] Ming Yuan and Cun-Hui Zhang. Incoherent tensor norms and their applications in higher order tensor completion. *IEEE Transactions on Information Theory*, 63(10):6753–6766, 2017.
- [67] Ali Zare, Alp Ozdemir, Mark A Iwen, and Selin Aviyente. Extension of pca to higher order data structures: An introduction to tensors, tensor decompositions, and tensor pca. *Proceedings of the IEEE*, 106(8):1341–1358, 2018.
- [68] Ali Zare, R. Wirth, C. Haselby, Heiko Hergert, and M. Iwen. Modewise johnson–lindenstrauss embeddings for nuclear many-body theory. *The European Physical Journal A*, 59, 05 2023.